

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



## THESIS

### WEB-ENABLING AN EARLY WARNING AND TRACKING SYSTEM FOR NETWORK VULNERABILITIES

by

James Wyatt Coffman

September 2001

Thesis Advisor:

Bert Lundy

Second Reader:

Roy M. Radcliffe

**Approved for public release; distribution is unlimited**

Report Documentation Page		
<b>Report Date</b> 30 Sep 2001	<b>Report Type</b> N/A	<b>Dates Covered (from... to)</b> -
<b>Title and Subtitle</b> Web-enabling an Early Warning and Tracking System for Network Vulnerabilities	<b>Contract Number</b>	
	<b>Grant Number</b>	
	<b>Program Element Number</b>	
<b>Author(s)</b> James Wyatt Coffman	<b>Project Number</b>	
	<b>Task Number</b>	
	<b>Work Unit Number</b>	
<b>Performing Organization Name(s) and Address(es)</b> Research Office Naval Postgraduate School Monterey, Ca 93943-5138	<b>Performing Organization Report Number</b>	
<b>Sponsoring/Monitoring Agency Name(s) and Address(es)</b>	<b>Sponsor/Monitor's Acronym(s)</b>	
	<b>Sponsor/Monitor's Report Number(s)</b>	
<b>Distribution/Availability Statement</b> Approved for public release, distribution unlimited		
<b>Supplementary Notes</b>		
<b>Abstract</b>		
<b>Subject Terms</b>		
<b>Report Classification</b> unclassified	<b>Classification of this page</b> unclassified	
<b>Classification of Abstract</b> unclassified	<b>Limitation of Abstract</b> UU	
<b>Number of Pages</b> 80		

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
<b>1. AGENCY USE ONLY (Leave blank)</b>		<b>2. REPORT DATE</b> September 2001	<b>3. REPORT TYPE AND DATES COVERED</b> Master's Thesis	
<b>4. TITLE AND SUBTITLE:</b> Web-enabling an Early Warning and Tracking System for Network Vulnerabilities			<b>5. FUNDING NUMBERS</b>	
<b>6. AUTHOR(S)</b> James Wyatt Coffman				
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b> Naval Postgraduate School Monterey, CA 93943-5000			<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> N/A			<b>10. SPONSORING / MONITORING AGENCY REPORT NUMBER</b>	
<b>11. SUPPLEMENTARY NOTES</b> The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
<b>12a. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited			<b>12b. DISTRIBUTION CODE</b>	
<b>13. ABSTRACT (maximum 200 words)</b> <p>The Information Assurance Vulnerability Alert (IAVA) process was established to provide an early warning and tracking capability for protecting Department of Defense (DoD) networks against identified system vulnerabilities. The Navy initially used record message traffic for the information distribution required by the process. This approach was heavily administrative and prone to significant delays in an already time critical process. Additionally, it lacked support for automated data validation, resulting in unreliable vulnerability tracking information. As a result, the process was ineffective, and Navy networks remained highly susceptible to exploitation, even for well-documented system vulnerabilities. For this thesis, web-enabling technology is used to build and deploy an early warning and tracking system for Navy network vulnerabilities. The research sponsor, the Navy Component Task Force for Computer Network Defense (NCTF-CND), has named it the Online Compliance Reporting System (OCRS). It is now being used by all Navy commands and has proven efficient and highly effective in defending Navy networks against known vulnerability exploitations. As a result, the system has gained significant interest from other organizations and the research sponsor is now planning to fund maintenance and future enhancements by the Space and Naval Warfare Systems Center in Charleston, South Carolina.</p>				
<b>14. SUBJECT TERMS</b> Computer Network Defense, CND, Early Warning and Tracking System, Information Assurance Vulnerability Alert, IAVA, Online Compliance Reporting System, OCRS, Web-enabled Information System, Network vulnerability notification, Network vulnerability reporting, Network vulnerability tracking, Distributed Administration, Hierarchical Administration, Web-enabled Navy			<b>15. NUMBER OF PAGES</b> 80	
			<b>16. PRICE CODE</b>	
<b>17. SECURITY CLASSIFICATION OF REPORT</b> Unclassified	<b>18. SECURITY CLASSIFICATION OF THIS PAGE</b> Unclassified	<b>19. SECURITY CLASSIFICATION OF ABSTRACT</b> Unclassified	<b>20. LIMITATION OF ABSTRACT</b> UL	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)  
Prescribed by ANSI Std. Z39-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**WEB-ENABLING AN EARLY WARNING AND TRACKING SYSTEM FOR  
NETWORK VULNERABILITIES**

James Wyatt Coffman  
Lieutenant Commander, United States Navy  
M.S., Information Technology Management, Naval Postgraduate School, 1998  
B.A., Computer Science, Rice University, 1989

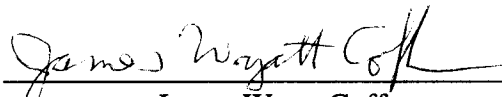
Submitted in partial fulfillment of the  
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

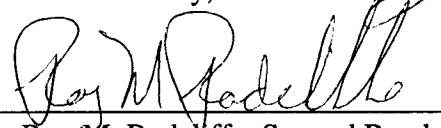
**NAVAL POSTGRADUATE SCHOOL  
September 2001**

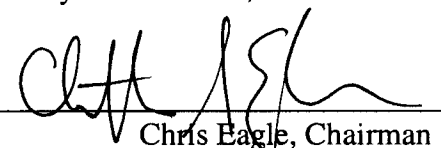
Author:

  
James Wyatt Coffman

Approved by:

  
Bert Lundy, Thesis Advisor

  
Roy M. Radcliffe, Second Reader

  
Chris Eagle, Chairman  
Information Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

## **ABSTRACT**

The Information Assurance Vulnerability Alert (IAVA) process was established to provide an early warning and tracking capability for protecting Department of Defense (DoD) networks against identified system vulnerabilities. The Navy initially used record message traffic for the information distribution required by the process. This approach was heavily administrative and prone to significant delays in an already time critical process. Additionally, it lacked support for automated data validation, resulting in unreliable vulnerability tracking information. As a result, the process was ineffective, and Navy networks remained highly susceptible to exploitation, even for well-documented system vulnerabilities. For this thesis, web-enabling technology is used to build and deploy an early warning and tracking system for Navy network vulnerabilities. The research sponsor, the Navy Component Task Force for Computer Network Defense (NCTF-CND), has named it the Online Compliance Reporting System (OCRS). It is now being used by all Navy commands and has proven efficient and highly effective in defending Navy networks against known vulnerability exploitations. As a result, the system has gained significant interest from other organizations and the research sponsor is now planning to fund maintenance and future enhancements by the Space and Naval Warfare Systems Center in Charleston, South Carolina.

THIS PAGE INTENTIONALLY LEFT BLANK



## TABLE OF CONTENTS

<b>I.</b>	<b>INTRODUCTION.....</b>	<b>1</b>
<b>A.</b>	<b>INFORMATION ASSURANCE VULNERABILITY ALERT PROCESS .....</b>	<b>1</b>
	1. Problems with the Navy’s Early Warning Phase.....	2
	2. Problems with the Navy’s Tracking Phase.....	3
	3. Impact of the Navy’s Initial Implementation.....	4
<b>B.</b>	<b>PROPOSAL: A WEB-BASED APPROACH TO THE PROCESS .....</b>	<b>5</b>
	1. Area of Research: Developing an Interactive Web Solution .....	5
	2. Resolving the Administrative Challenges of a Web-enabled Solution .....	6
<b>C.</b>	<b>SUMMARY OF REMAINING CHAPTERS.....</b>	<b>7</b>
<b>II.</b>	<b>IDENTIFYING REQUIREMENTS AND SELECTING WEB TECHNOLOGY .....</b>	<b>9</b>
<b>A.</b>	<b>IDENTIFYING REQUIREMENTS FOR A WEB-ENABLED SOLUTION .....</b>	<b>9</b>
	1. System User Requirements .....	9
	a. <i>Network Action Officers .....</i>	<i>9</i>
	b. <i>System Administrators .....</i>	<i>10</i>
	2. Vulnerability Warning and Tracking Process Require ments .....	10
	a. <i>Providing an Early Warning for Network Vulnerabilities....</i>	<i>10</i>
	b. <i>Collecting Network Vulnerability Compliance Reports .....</i>	<i>11</i>
	c. <i>Tracking the Status of Vulnerable Network Systems .....</i>	<i>12</i>
	3. Web-Based Administration Requirements .....	12
	a. <i>Registering for a User Account .....</i>	<i>13</i>
	b. <i>Logging In and Out.....</i>	<i>15</i>
	c. <i>Adding Subordinate Commands.....</i>	<i>15</i>
	d. <i>Approving Subordinate User Accounts.....</i>	<i>16</i>
	e. <i>Managing Passwords .....</i>	<i>17</i>
	f. <i>Modifying Personal Account Information.....</i>	<i>18</i>
	g. <i>Closing User Accounts.....</i>	<i>18</i>
	h. <i>Changing the Organizational Hierarchy .....</i>	<i>18</i>
	i. <i>Generating a Current Mailing List of Active Users.....</i>	<i>19</i>
<b>B.</b>	<b>SELECTING TECHNOLOGY FOR A WEB-ENABLED SOLUTION..</b>	<b>19</b>
	1. Choosing a Server to Host the Web Solution .....	20
	2. Picking a Database to Store Persistent Data for Web Applications .....	20
	3. Selecting Web Application Development and Hosting Software...	20
<b>C.</b>	<b>SUMMARY .....</b>	<b>21</b>
<b>III.</b>	<b>THE ONLINE COMPLIANCE REPORTING SYSTEM .....</b>	<b>23</b>
<b>A.</b>	<b>USER INTERACTION WITH WEB-BASED SERVERS .....</b>	<b>23</b>

1.	Requesting Static Web Pages .....	25
2.	Interacting with Web Applications .....	25
B.	KEY WEB APPLICATIONS .....	26
1.	Administrative Applications .....	27
a.	Authenticating User Access .....	27
b.	Registering for a New Account.....	28
c.	Building a Subordinate Organizational Hierarchy .....	31
d.	Approving Pending Subordinate Account Requests.....	33
e.	Other Administrative Applications.....	35
2.	Information Assurance Vulnerability Alert Applications .....	36
a.	Early Warning.....	36
b.	Compliance Reporting .....	38
c.	Vulnerability Tracking.....	39
C.	SUMMARY .....	41
IV.	SECURITY, LESSONS LEARNED, AND FUTURE ENHANCEMENTS .....	43
A.	SECURITY .....	43
1.	Internet Domain Restrictions .....	44
2.	Encrypted Web Server Connections .....	44
3.	Password Format Validation .....	44
4.	Ensuring Password Confidentiality .....	45
5.	Auto-Expiring User Session Credentials .....	45
6.	Account Hijacking Protection.....	46
7.	Automated Password Retrieval .....	46
8.	Database Security.....	48
B.	LESSONS LEARNED FROM USER INTERACTION .....	48
1.	Training.....	49
2.	Network Access .....	50
C.	PROPOSALS FOR FUTURE SYSTEM ENHANCEMENTS .....	50
1.	Additional Reporting Capabilities.....	51
2.	Acknowledging Compliance Requirements .....	51
3.	Consolidated Compliance Status Displays.....	52
4.	Variable Access Privileges.....	52
5.	Mandatory Periodic Password Changes.....	53
D.	SUMMARY .....	53
V.	IMPACT ON THE NAVY .....	55
A.	THE NAVY'S PROGRESS USING A WEB-BASED APPROACH .....	55
1.	Navy Chooses from Two Potential Web Solutions .....	55
2.	Web-based System is Available to Every Command in the Navy.....	56
3.	New System Requires Minimal Administrative Support .....	57
4.	System Flexibility Supports Addition of New Tasking Order Capability .....	57
5.	Code Red: A Navy Network Defense Success Story .....	57
6.	Web System Garners Positive Feedback and Navy-wide Awareness.....	58

7.	Successful System Generates Interest From Other Organizations .....	59
B.	CRITICAL FACTOR: SHIFTING THE BURDEN OF RESPONSIBILITY.....	59
C.	SUMMARY.....	60
VI.	CONCLUSIONS .....	61
A.	WEB-ENABLED EARLY WARNING .....	61
B.	WEB-BASED NETWORK VULNERABILITY TRACKING .....	62
C.	DISTRIBUTED ADMINISTRATION FOR A WEB SOLUTION .....	63
D.	CONCLUSION .....	63
	BIBLIOGRAPHY .....	65
	INITIAL DISTRIBUTION LIST .....	67

THIS PAGE INTENTIONALLY LEFT BLANK

## LIST OF FIGURES

Figure 1.	The Web-enabled Vulnerability Process.....	11
Figure 2.	Web-based Administration Capabilities. ....	14
Figure 3.	Client-Server Data Flow. ....	24
Figure 4.	Login Screen. ....	27
Figure 5.	New Account Registration Form. ....	29
Figure 6.	Command Information Screen. ....	32
Figure 7.	New Command Form.....	33
Figure 8.	Subordinate Account Approval Form. ....	35
Figure 9.	New Vulnerability Form. ....	37
Figure 10.	Vulnerability Compliance Report Form. ....	39
Figure 11.	Status Tracking Screen. ....	41

THIS PAGE INTENTIONALLY LEFT BLANK

## **I. INTRODUCTION**

This thesis continues previous work done by the author to establish the Information Assurance Vulnerability Alert (IAVA) dissemination and reporting process within the Navy. This chapter describes the process and then, based on the author's first-hand experience, details the problems with the Navy's initial implementation. Next, a web-based approach to resolving the original problems is discussed, which leads to a preview of the solution that was developed during this research. And finally, a brief overview of the remaining chapters is provided.

### **A. INFORMATION ASSURANCE VULNERABILITY ALERT PROCESS**

The Information Assurance Vulnerability Alert process was established in June 1998. It was done in response to Assistant Secretary of Defense direction that all Department of Defense (DoD) Commanders in Chief (CINCs), Services, and Agencies establish a dissemination and compliance tracking capability for protecting defense networks against system vulnerabilities. The process is primarily intended to protect against well-known vulnerabilities, like network system buffer overflows, that are prone to exploitation by third party threats, especially hackers. As the designated agent for the process, the Defense Information Systems Agency (DISA) is responsible for issuing vulnerability messages to all services and agencies. This begins the early warning phase of the process, alerting network action officers to a potential vulnerability with installed network systems. Each of the services and agencies is then responsible for their own internal distribution of vulnerability messages. The Navy Component Task Force for Computer Network Defense (NCTF-CND) is responsible for internal notification throughout the Navy.

Each vulnerability message contains procedures for securing systems against an identified vulnerability. When a vulnerability message is issued, organizations are required to inventory their vulnerable systems and to follow the prescribed procedures to protect those systems that are affected. Individual commands are then responsible for submitting reports detailing their vulnerability compliance status. This starts the tracking phase of the process, for monitoring the status of vulnerable network systems. Compliance reports are required to include vulnerable system inventories as categorized

by total number of systems, number of affected systems, number of corrected systems, and the number waived due to special circumstances. For the Navy, the reports are submitted, via the appropriate chain of command, to the Navy Component Task Force for Computer Network Defense. They, in turn, compile the data contained in the individual reports and submit a Navy-wide compliance status report to the Defense Information Systems Agency. The entire notification and reporting process must be completed, with vulnerabilities corrected and reported, within a specified time frame, usually 30 days.

### **1. Problems with the Navy's Early Warning Phase**

Vulnerability notification is a time critical function because there are only 30 days to complete the reporting process for the entire Navy. The fundamental objective is to get known vulnerability information into the hands of appropriate network action officers as quickly as possible. Doing so provides the greatest opportunity to take action and defend against potential exploitation. One of the major challenges for the Navy is to quickly distribute vulnerability messages to network action officers at all commands. This has proven to be particularly difficult due to the Navy's organizational structure and the exceptionally large number of commands.

The primary method used to distribute administrative information throughout the Navy (and other military services) is record message traffic. It is essentially a wire service that uses a standard text-based format and provides a global distribution system. In recent years, the addition of networks at most Navy commands has improved local record message traffic distribution. Incoming messages are now typically routed via electronic mail to appropriate action officers rather than being printed and routed manually.

When the Information Assurance Vulnerability Alert process was first implemented in late 1998, the Navy used record message traffic to distribute vulnerability notices internally. However, there were two problems with the use of message traffic that added undesirable delays. First, the Navy is a hierarchical organization and messages were routed down the chain of command, from superior to subordinate. This added delays at each level, as the messages had to be re-addressed to action officers at subordinate levels. Second, within a command or organization, record message traffic is routinely addressed for action at the department head level and above. This added



additional delay since the messages had to be read and then routed down to the appropriate network action officers within the departments at each command. Thus, the use of record message traffic as a vulnerability message distribution method could take several days or longer, especially for commands at the bottom of the Navy's organizational hierarchy.

## **2. Problems with the Navy's Tracking Phase**

Like the notification process, vulnerability compliance reporting is a time critical function, but it also carries the burden of accuracy and accountability for the data being reported. The fundamental goal is to depict an organization's current level of readiness for defense against vulnerability exploitation. This provides decision-makers with the ability to identify and track areas of strengths and weaknesses within their networks on a command-by-command basis. The Navy's challenge is to quickly collect vulnerability compliance reports from all commands and then to generate and submit a consolidated report that depicts an accurate view of the Navy's status. This has proven almost impossible to accomplish accurately and efficiently using traditional collection methods due to the Navy's organizational structure and the large volume of reports that has to be processed.

The Navy also used record message traffic as a means to collect the vulnerability compliance reports via the appropriate chain of command. Each command was required to submit a status report to their superior that also included the compliance status for all of their subordinate commands. This process suffered from a variety of problems that added delay, reduced accuracy and essentially eliminated accountability on a command-by-command basis.

First, a significant delay resulted from submitting reports via the chain of command. In addition to the hierarchical delay caused by traversing the chain of command one level at a time, each command was also limited by the efficiency of their slowest subordinate. A consolidated report could not be generated and forwarded until all subordinate reports had been received. This resulted in some major delays, especially in those cases where there was a deep organizational hierarchy.

Next, there was the problem of report format. Despite having a standard format specified in each vulnerability-warning message, most commands would submit data in an undesirable format. For example, blank fields were skipped or deleted when a zero was required if no inventory was available. In many cases, explanatory comments were entered in each field next to the data instead of at the end in a comment block. Some reports were submitted one field per line and others contained multiple fields per line. Overall, the format problem went unchecked because the reports were submitted via text-based messages that did not support field or format validation. As a result, there was no way to automatically tabulate the data in the compliance reports. So, at each level in the command hierarchy, an additional reporting delay was incurred due to the necessity for manually tabulating data received in each subordinate report.

Additionally, in many cases the data was difficult to decipher from the reports. This resulted in assumptions or guesses as to what was intended, which significantly reduced the accuracy of the data being reported. Furthermore, in an effort to ensure reports from all subordinate commands were included, there was a strong tendency to use the format with the lowest common denominator. By the time the reports made their way up the entire chain of command, the consolidated reports consisted of little more than percentage estimates regarding compliance status. For example, a consolidated report for an entire subordinate hierarchy might state that 82% of the vulnerable systems were in compliance. This resulted in a complete loss of accountability with respect to the status and inventory levels on a command-by-command basis.

### **3. Impact of the Navy's Initial Implementation**

Due to the problems described above, the Navy's vulnerability notification and reporting processes were failing to achieve their desired objectives. The early warning system was ineffective since Navy network action officers were not receiving vulnerability information in a timely fashion. The compliance status reports were not sufficient to depict the Navy's level of readiness for defense against vulnerability exploitation. And there was no way for decision-makers to identify and track strengths and weaknesses in the networks on a command-by-command basis. Additionally, due to all of the delays incurred, the Navy was unable to complete the process in the required time for any of the vulnerabilities issued using record message traffic as the primary

means for information dissemination and retrieval. As a result, Navy networks remained highly susceptible to exploitation, even for well-documented system vulnerabilities. Unless a better solution was implemented, the Navy would be unable to meet Department of Defense requirements for the Information Assurance Vulnerability Alert process.

## **B. PROPOSAL: A WEB-BASED APPROACH TO THE PROCESS**

Most Navy activities are using web sites to disseminate information. Following this trend, a web site could be used by the host organization to post vulnerability-warning messages. However, a simple web site cannot meet the vulnerability notification requirements because the network action officers at each command still need to be notified when a network vulnerability is posted before they can take action. To some extent, electronic mail (email) can support in this regard, but without knowing exactly who to send messages to, an organizational hierarchy will still have to be traversed in order to get the information to the appropriate network action officers. Maintaining an up-to-date mailing list for thousands of action officers in a large and dynamic organization is no easy task. Furthermore, even with an accurate list of action officers and their email addresses, the problems associated with forwarding and consolidating compliance reports will persist without some additional automated information collection and processing capabilities. Thus, some type of dynamic online capability is required to fully address these concerns.

### **1. Area of Research: Developing an Interactive Web Solution**

The major objective of this research was to design and build a web-based system capable of efficiently and effectively handling the vulnerability notification and reporting functions for all commands by specifically addressing the timeliness, accuracy, and accountability concerns with the Navy's existing implementation. There were three goals: (1) disseminate vulnerability notices directly to network action officers as quickly as possible; (2) collect compliance reports, then automatically summarize and forward the data; and (3) provide a secure online environment for managing the entire process. The scope of this research included the use of a persistent data store and the development of numerous interactive web-based applications that could provide automated on-line data collection, with dynamic tabulation and real-time status display. It ultimately required the integration of a web server, a database server, and an application server.

The end result of this thesis is a web-enabled early warning and tracking system for network vulnerabilities. A prototype with minimal capabilities was brought online in September 1999. The system has been modified and rewritten several times over the last year and a half and is now relatively mature and available for use throughout the Navy. It has been named the Online Compliance Reporting System (OCRS) by its host organization, the Navy Component Task Force for Computer Network Defense (NCTF-CND). The purpose of the system is to quickly disseminate vulnerability warnings directly to all network action officers and then to collect and track the vulnerability compliance reports from each Navy command. The system automatically organizes, summarizes and presents the data according to the appropriate chain of command. The hierarchical structure of the system provides each account holder with a real-time summary view of their entire subordinate organizational hierarchy and further allows them to drill down and review the individual vulnerability compliance status of any subordinate command. Furthermore, it automatically prepares the consolidated Navy report, which the host organization is required to submit to the Defense Information Systems Agency. The web-based system will be described in greater detail throughout the remainder of this thesis.

## **2. Resolving the Administrative Challenges of a Web-enabled Solution**

In developing a web-based solution, one of the major challenges was to minimize administrative support requirements (account registration and approval, password management, etc.). This was especially important considering there would be thousands of account holders accessing and using it from Navy activities all over the world. To overcome this problem, account administration was decentralized by distributing key functions down to all account holders. This was done without jeopardizing the security of the system or unnecessarily burdening subordinate account holders with complex administrative requirements. The distributed functions were incorporated in a hierarchical fashion. This provides each account holder with the full authority to administer all accounts directly subordinate to them in their chain of command. The administrative functions that are distributed in this manner will also be discussed in this thesis.

## **C. SUMMARY OF REMAINING CHAPTERS**

The remainder of this thesis will cover the Online Compliance Reporting System in much greater detail. Chapter II identifies and discusses the user requirements, the vulnerability process requirements, and the web-based administration requirements that define the structural objectives for developing the web-based applications. It also describes the selection and integration of web resources used to build and support the system. Chapter III walks the reader through the system implementation. It includes descriptions of the key web applications that were developed and displays several screenshots. Security features and recommendations are discussed in Chapter IV, along with lessons learned and specific proposals for future enhancements. Chapter V discusses the impact the web-based system has had on the Navy and on the network vulnerability notification and reporting process. It also discusses the issue of distributed administrative responsibility, considered by the author to be one of the primary keys to success. And finally, Chapter VI provides conclusions based on this research.

THIS PAGE INTENTIONALLY LEFT BLANK

## **II. IDENTIFYING REQUIREMENTS AND SELECTING WEB TECHNOLOGY**

This chapter discusses the requirements and technology that went into the design and development of the Online Compliance Reporting System (OCRS). The full range of requirements is presented. These include the user requirements, the Information Assurance Vulnerability Alert (IAVA) process requirements, and the administration requirements that are critical to the web-based approach. Additionally, a brief discussion is included on the selection and integration of the server technology required for the web-based system.

### **A. IDENTIFYING REQUIREMENTS FOR A WEB-ENABLED SOLUTION**

This section discusses the user requirements, the process requirements, and the web-based administration requirements that have been identified and are supported in the current version of the system. In addition to analysis of the vulnerability warning and tracking process, a build-and-fix development approach was used to derive requirements based on user feedback.

#### **1. System User Requirements**

User requirements determine the online functions that are needed to fully automate the process via the web. Throughout the course of the development effort, two categories of clients, or user groups, are identified: network action officers and system administrators. Each user group requires different access levels and specific types of interaction with the system. Several different web applications will be required to support the different capabilities.

##### ***a. Network Action Officers***

Network action officers represent the primary user group (well over 2000). Every user has network action officer privileges and responsibilities, including the system administrators discussed next. These responsibilities include taking action on posted vulnerability messages and reporting compliance results as directed. Therefore, action officers require access to the system in order to read posted warning messages and to submit compliance reports for their commands. Those with subordinate activities require a capability to track the reporting status of their subordinates. As a result, they will also

be required to identify their subordinate activities to the system. Furthermore, network action officers will be responsible for managing the accounts of the action officers from their subordinate activities.

***b. System Administrators***

The top, or root level, organization in the system is the central authority for the vulnerability warning and tracking process. The Navy Component Task Force for Computer Network Defense (NCTF-CND) fills this role for the Navy. Account holders from this organization are called system administrators. In addition to network action officer requirements, they also have other special requirements that are essential to maintaining the system. They require a different level of access in order to post new vulnerability messages, to notify all network action officers of newly posted messages, and to change the Navy's organizational hierarchy (as needed). They are also responsible for the setup, hosting and maintenance of the system. Additionally, they are responsible for handling trouble calls and providing support and/or training to network action officers if necessary.

**2. Vulnerability Warning and Tracking Process Requirements**

Three primary functions related to the vulnerability warning and tracking process are identified in support of the user requirements described above. Each of these functional requirements results in one or more web-enabled applications being developed for the system and then made available for use over the Internet. The block diagram in **Figure 1** shows the Online Compliance Reporting System (OCRS) inputs and outputs that are based on the following three primary vulnerability process requirements:

- Provide an early warning for network vulnerabilities
- Collect network vulnerability compliance reports
- Track the status of vulnerable network systems

***a. Providing an Early Warning for Network Vulnerabilities***

To support web-based early warning notification process, three supporting requirements are identified. First, system administrators require an application for posting new vulnerability messages to the system. The application also needs the capability to later modify the posted messages if needed. To dynamically track each network vulnerability posted to the system, the message, along with a short vulnerability



description, a tracking number, and due date information, needs to be stored in an on-line database, along with the poster's account number (for accountability). Next, a web application is required that is capable of identifying and directly notifying (via email) all network action officers (with active accounts) when a new vulnerability message is posted to the system. And finally, another application is needed to provide all network action officers with on-line access to the posted vulnerability messages. The inputs and outputs, labeled A, B, and C in **Figure 1**, are associated with these supporting requirements.

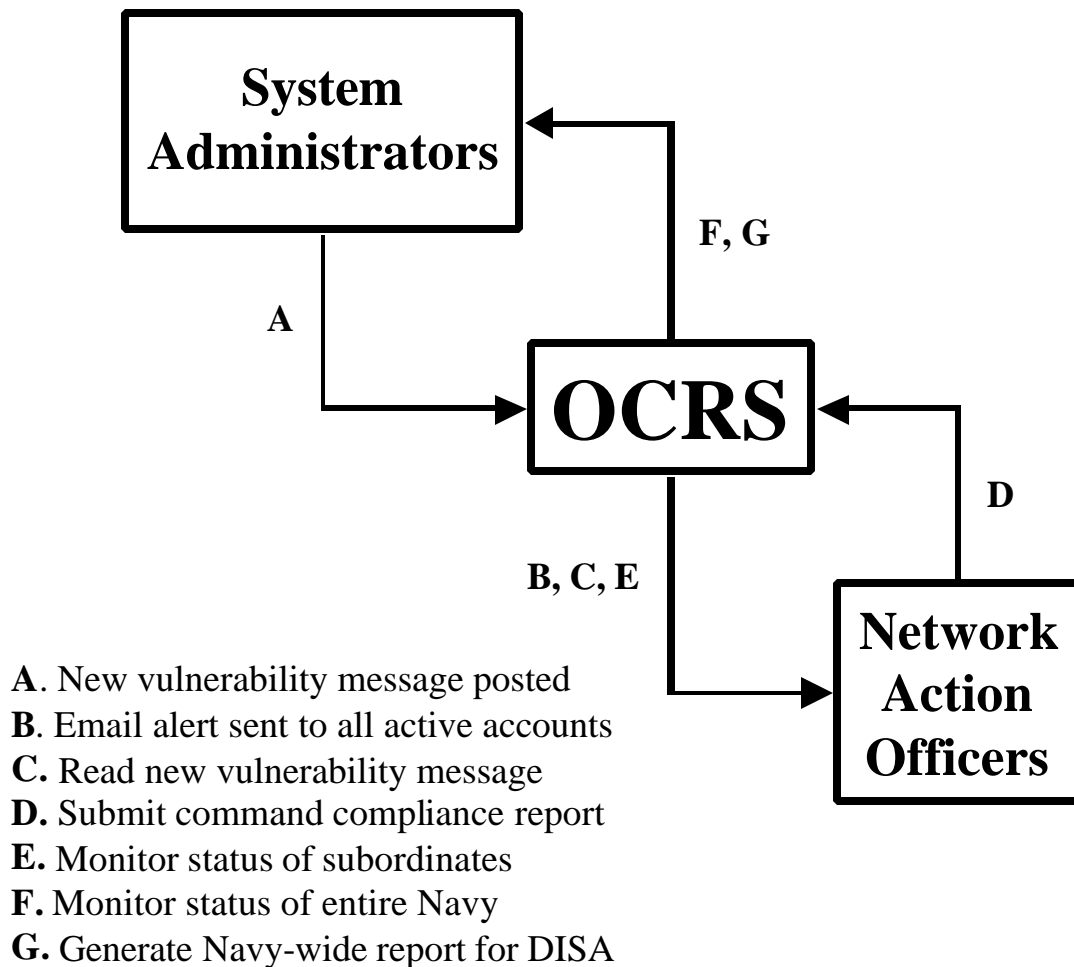


Figure 1. The Web-enabled Vulnerability Process.

***b. Collecting Network Vulnerability Compliance Reports***

Collecting vulnerability compliance reports from all commands requires an application with the capability to present and process a standardized web-based form.

Input D (from **Figure 1**) represents the requirement for this application. The application needs to auto-validate the reports submitted to the greatest extent possible in order to prevent errors in the data collected. If errors are identified, the application needs to reject the report and redisplay the form, including the incorrect data and an appropriate error message. Each compliance report needs to be automatically stored in an on-line database along with the submission date and the network action officer's account number (for accountability). The application must first present an action officer with a previously submitted report, instead of a blank report form, if a report has already been submitted for the organization in question. And, there must be an option to modify previously submitted reports. Additionally, there must be a capability for a network action officer to submit a report for any subordinate command in the event subordinate action officers cannot access the system for any reason. The system is required to ensure no more than one report is submitted for each organization per vulnerability message.

*c.      **Tracking the Status of Vulnerable Network Systems***

Consolidating compliance reports from all commands requires an application with the capability to dynamically (real-time) generate a status, or summary, based on all compliance reports received for a given vulnerability message. The inputs and outputs, labeled E, F, and G in **Figure 1**, show the supporting requirements for this application. The application needs to summarize and present compliance data, but only for the current network action officer's command and all of its subordinate commands. Furthermore, the status application needs to provide a capability for an action officer to drill-down to any subordinate level, to display a subordinate summary view, or even to display a specific subordinate organization's actual compliance report. Subordinate commands with missing compliance reports need to be clearly identified in the web-based summary report. For the system administrators, this application will be used to generate the Navy-wide compliance report that will be submitted to the Defense Information Systems Agency (DISA).

**3.      Web-Based Administration Requirements**

Several administrative functions that are critical to the support of a web-enabled process are identified during early development. Each of these functional requirements results in one or more web applications being developed and incorporated into the

system. The block diagram in **Figure 2** shows the system inputs and outputs that are based on the following administration requirements. Each of these is discussed in greater detail below:

- Registering for a new user account
- Logging in and out of the system
- Adding subordinate commands to the system database
- Approving pending subordinate user accounts
- Managing passwords
- Modifying personal account information
- Closing user accounts
- Changing the Navy organizational hierarchy
- Generating a current mailing list of active users

*a. Registering for a User Account*

Every network action officer is required to register for an account using the online system. In order to reduce administration, registering for an account is required to actually establish the user account in the database with a status of “pending approval.” This prevents the account user from accessing the system until approved. Use of an on-line web form is required to register for an account. This form will be available from a button on the login screen. The application that presents and processes this form is required to automatically validate as much information from the form as possible. Mandatory fields must be supplied, and verification is required. The form contains a field labeled as the Unit Identification Code (UIC). This field is used to link the user to a command that must already be in the system’s database. Also, registering network action officers are required to make up their own user names and passwords. To enhance security, passwords must follow strict rules that the registration application can verify. Furthermore, the application will check the existing database to ensure the user doesn’t already have an account. If errors or problems are identified in the input, the application will redisplay the form with the bad data and provide clear and specific guidance for making corrections.

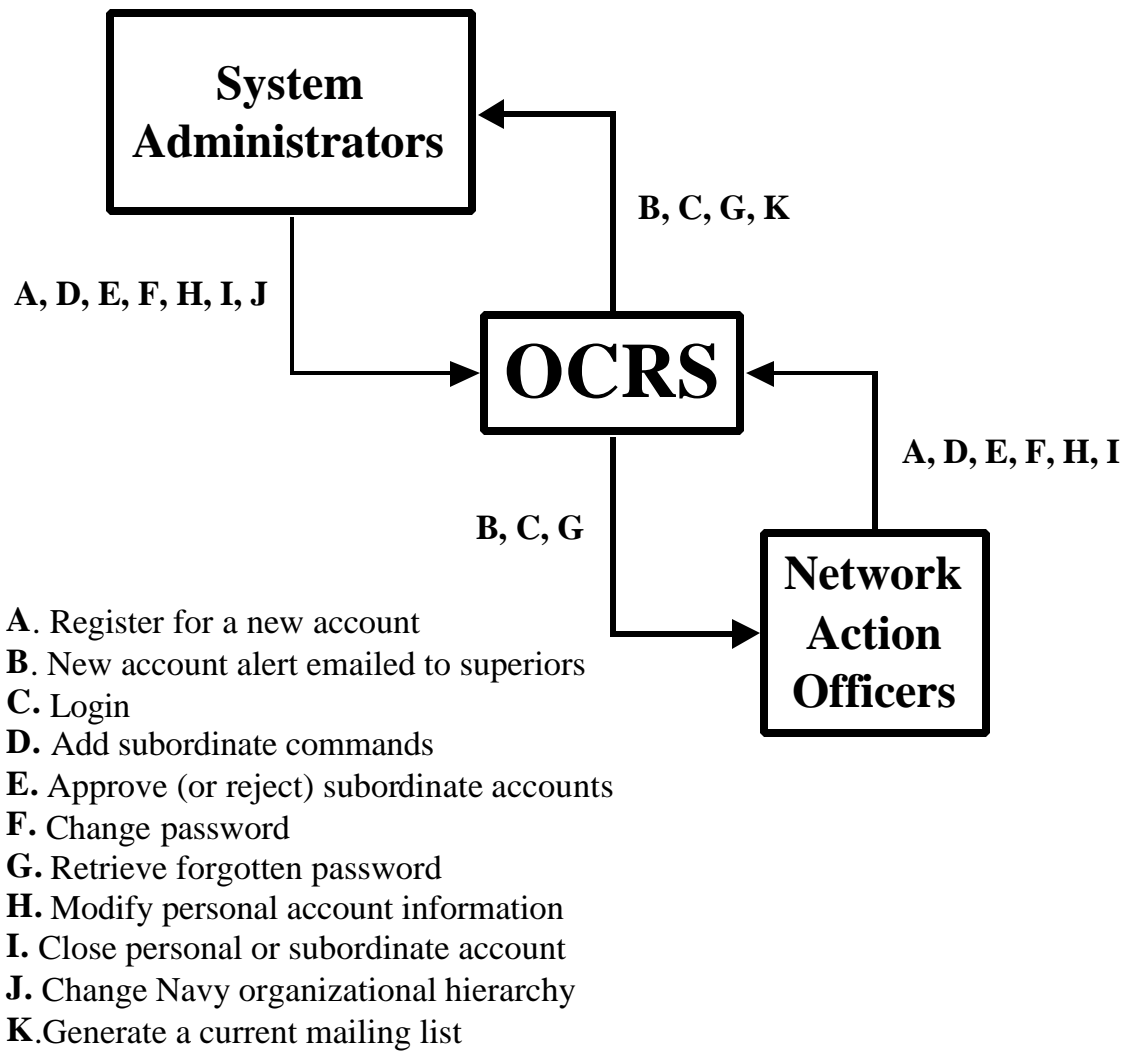


Figure 2. Web-based Administration Capabilities.

Once the application accepts the data as valid (to the degree that can be automatically verified), a confirmation screen will be displayed to the account requestor, showing the actual name of the requestor's command, along with the other data submitted. The requestor will then have the option of modifying or confirming the request. If the request is confirmed, the application will automatically send an email message to each account holder at the requestor's superior command. The email message will notify all of the appropriate action officers that a subordinate account request has been submitted and requires approval (discussed below). Finally, the application will inform the requestor that the new account request has been received, the superior

administrators have been notified, and the account is pending final approval with notification to follow.

***b. Logging In and Out***

A login application is necessary in order to provide access to the system for users with approved accounts. Its primary purpose is to restrict access to authorized users. When used to gain access to the system, this application will provide a welcome screen that displays all functions for which the user has access. This means that users and applications will be assigned access levels to support a run-time comparison by the login application. The login application will also be required to take the user to the new password screen if the user's password is flagged for expiration. Additionally, if the user's password has not been flagged for expiration, and there are subordinate accounts pending approval, the login application will be required to display all pending subordinate accounts in order to allow the user to approve or reject those accounts. Finally, if the user's password is not expired, and there are no pending subordinate accounts, then the login application will need to display a list of the ten most recently posted vulnerability messages. A button, which can be used to log out of the system, will be required on all screens while logged into the system. If there is no user activity for fifteen minutes, the user will be automatically logged out of the system.

***c. Adding Subordinate Commands***

Since the Online Compliance Reporting System is designed to be a hierarchical system, a database containing hierarchical relationship information is required. Because of the need to minimize administrative staff support and to have an accurate representation of the current Navy hierarchy, no organizational data will be identified or preloaded into the database. Instead, only the top-level organization and the initial system administrator will be preloaded. This will allow root level users, or system administrators, to register for system access. When approved, users will then be required to enter a list of commands that are immediately subordinate to their own (if the list is not already present). Any user at a command may do this for their subordinate commands. Once a subordinate command is added to the system in this manner, users at that command may then register for an account, and then repeat the process for their own subordinates when approved.

This recursive approach will allow for a more accurate organizational hierarchy while also minimizing the root level administrative staff by distributing the burden of responsibility to network action officers for their own subordinates. Therefore, an application will be required to allow all network action officers to add and/or delete their subordinate commands to/from the system. The identity of an action officer who adds or modifies a subordinate command will be stored with the command's record. This will provide user accountability for the organizational hierarchy stored within the database. This application will also be required to disallow the removal of a subordinate command from the database if there are users and/or vulnerability compliance reports already associated with that command.

*d. Approving Subordinate User Accounts*

Approving subordinate accounts will require a confirmation process to ensure that each user requesting an account is authorized to have access to the system. This would be a tedious and time-consuming process for a central staff, especially with thousands of account requests coming in from Navy network action officers all over the world. To eliminate the central staff requirements, and to eliminate long delays in the approval process, this function will also be distributed down to subordinate users. Thus each network action officer will be required to approve and/or reject new account requests from users in commands that are immediately subordinate to their own. By doing this, the burden of approval is moved much closer to the users that are requesting accounts, thereby ensuring the approval authority will be better acquainted with the requestors (i.e., more likely to know them and their authorization status). Also, the thousands of account requests will be more evenly distributed across the entire user base, which will significantly minimize response times as well as central administrative staff requirements.

As described previously, when a new user account request is made, the users for the superior command will be immediately notified by email that a subordinate account request is pending approval. When a network action officer logs in after receiving one of these notices, he/she will be immediately taken to the approval application, which will list all pending subordinate account requests. The user will then be allowed to select an account to approve (or reject) and will be taken to the account

validation screen. Thus, to ensure the approval process is standardized across the entire organization, the online approval application will require a standard form. The form will need to be a checklist that can be completed by an approver as he/she goes through the process of validating an account request. When the checklist is completed and the approver presses the approve button, the account will be changed to an “active status” and the approver identity will be stored with the approved account record for accountability. Then, an email message will be automatically sent to notify the account user that the account has been approved.

In addition to the approval checklist and approve button, the form will display all of the personal information for the requestor. There will also be a text field with a preformatted email message for use when a rejection is required. The message may be modified to reflect why the request is being rejected before pressing the reject button, which will send the message, via email, to the requestor and change the account to a “rejected status.”

*e. Managing Passwords*

Password management would normally be done entirely by a central staff, but one of the requirements for this system is minimal system administrator support. To achieve this, functions necessary to manage passwords will be distributed entirely to each individual user. First, as described previously, each user will be required to create their own username and password when they register for an account. Second, a password modification application will be provided to allow each user to change their password whenever they like (or are required to). This application will require their old password (to prevent a passerby from hijacking their account) and will enforce the strict password generation rules used in the registration application to ensure secure passwords are always used. And finally, some means of recovering a lost or forgotten password will be required. To support this, a third application, which will be accessible from the login screen, will allow a user to retrieve a password by entering their email address in an online form. If their email address matches that of an account in the system, an email with the password will be sent to the address and the password will be automatically expired. Then, since email messages are readable if intercepted, when the user logs in,

he/she will be immediately taken to the password modification application to change the password before proceeding to other system applications.

***f. Modifying Personal Account Information***

To support changing user information, the system will be required to provide an application that allows users to modify their personal contact information. By distributing this capability, all users will be able to change their title (i.e., from Miss to Mrs.), email address, and/or phone numbers as needed, thereby alleviating the central staff of this responsibility. In order to maintain full accountability within the system, the application will not allow users to change their real names or their usernames. For security, a password will be required when submitting the online form to make any changes.

***g. Closing User Accounts***

In the Navy, personnel transfer from one activity to another fairly often, especially active-duty personnel. As a result, a means to easily close accounts for transferring personnel will be essential. This will be accomplished in two ways. First, an application will be provided that will allow each user to close his/her own account. To do this will require a user to enter a password. If the online form for closing a user account is submitted by that user, the user's account will be changed to a "closed status," and the user will be immediately logged out. The second way to close an account will be via an application that allows any user to close the account of any subordinate user (but not peer accounts) at any level in the subordinate hierarchy. If this method is used, the account being closed will also be changed to a "closed status" and will then be reflected as such in subordinate account listings. Also, the identity of the user closing the account will be logged with the closed account. Once accounts are closed, their users will no longer be able to log into the system. The accounts will not be deleted because their users might have submitted reports or entered other information into the system. By keeping the accounts in the database in a closed status, the accountability information (i.e., who did what and when) will always be available.

***h. Changing the Organizational Hierarchy***

The Online Compliance Reporting System will also need to support a dynamic organizational hierarchy. In an organization as large as the U.S. Navy,



subordinate commands and activities occasionally shift within the hierarchy, which creates a new chain of command for the shifting activity, as well as all subordinates to that activity. As a result, an application will be needed that can easily modify the organizational hierarchy on the fly. This application will need to change the superior identifier for a command that is being relocated. Furthermore, it will have to change the chain of command for that particular command and all of its subordinates. This application will only be available to the system administrators (central staff) at the root level organization because commands might be required to move between two different subordinate hierarchies, both of which will not be accessible by the subordinate users involved in the organizational change. Furthermore, this application will have to clearly identify potential changes that will result to the organizational hierarchy (before actually processing a change request). This will allow the system administrator to verify the consequences prior to proceeding.

*i. Generating a Current Mailing List of Active Users*

In addition to warning notices after posting new vulnerability messages to the system, the system administrators will also need to notify all account users of changes to the system, including upgrades and/or potential down time. As a result, an application will be required to dynamically generate an email mailing list of all currently active user accounts. This application will only be available to system administrators, who may use the list that is generated to formulate mass electronic mailings to all active account holders as deemed necessary.

**B. SELECTING TECHNOLOGY FOR A WEB-ENABLED SOLUTION**

A variety of hardware and software technologies are necessary to develop and use a fully web-capable early warning and tracking system. This section briefly discusses the selection of the primary server resources that are integrated to provide the full suite of web capabilities. To provide the dynamic capability required by web applications supporting the vulnerability notification and reporting process, and by the administrative applications described above, the following three server components are required. Each is discussed further below:

- A web server
- A database server

- An application server

### **1. Choosing a Server to Host the Web Solution**

Navy standards require the use of a Microsoft Windows NT based server for deploying a web-enabled system. Additionally, the web server is required to have built-in support for the Secure Socket Layer protocol, version three (SSL3), which allows for encrypted exchange of information between the web server and its browser-based clients. There are basically two choices that meet these conditions and carry no additional costs, Microsoft's Internet Information Server (IIS), which comes with Microsoft's Windows NT Server, and Netscape Communication's Enterprise Server, for which there is a free government license. The Netscape Enterprise Server was selected as the host web server because the Microsoft web server is not compatible with the Navy's security certificates, which are required to support the encrypted SSL3 connections.

### **2. Picking a Database to Store Persistent Data for Web Applications**

To provide a web-based solution that collects, retrieves and manipulates data for dynamic status reports from a large number of client commands in real-time requires the use of a database. A database server that is Open Database Connectivity (ODBC) compliant can be plugged into most commercial application servers and used as a backend data store for web-based applications. Furthermore, if the database is simply used as a repository, with no embedded business logic, the database server can be swapped out with just about any other database server should a higher backend storage capability be required. Thus, with Open Database Connectivity as a requirement, and using cost-avoidance as a selection principle, Microsoft Access database software was chosen because it comes pre-packaged with Microsoft Office, and it is supported directly by the Windows NT Server software. Microsoft Access is more than adequate for the initial use of the system, and it can easily be changed to a higher end solution, like Microsoft's SQL Server, should the need arise.

### **3. Selecting Web Application Development and Hosting Software**

An application server is needed to enable direct interaction between the web server and the database. Web applications developed for the server will provide all of the business logic necessary to allow for automating and managing the vulnerability process dynamically via the web. Given the above web server and database requirements, the

application server will need to run on a Windows NT Server, work with Netscape's Enterprise Server, and be able to connect with a Microsoft Access database. Everyware Development Corporation's Tango Enterprise software was selected because it meets the above requirements. Additionally, it is relatively inexpensive, and it is a fully cross-platform compatible solution, which can be used for application development on Windows and Macintosh systems and for deployment on Windows NT, Macintosh, or Unix systems. It should be noted that the Tango Enterprise software has been sold twice since this project began, first to Pervasive Corporation and then, more recently, to With Holding Corporation. It is now called WiTango, but will be called Tango throughout the remainder of this document.

The Tango software consists of two applications: Tango Editor and Tango Server. Tango Editor is used to develop web applications. It runs on Microsoft Windows and Apple Macintosh platforms. All of the web application development for this thesis was completed on an Apple Macintosh system and then transferred to a Windows NT server for deployment. Tango Server is a web server plug-in (or application server) that extends the capability of a web server to process the web applications developed with Tango Editor. The Tango Server plug-in works with most web servers that run on Microsoft Windows NT, Apple Macintosh and Unix systems. It runs on the same machine as the web server. Additionally, if Microsoft Access is used with the Tango Server, the Access database must also reside on the same machine. This requirement does not extend to other Open Database Connectivity databases, which may be hosted on a separate machine. For cost purposes, a single machine is used for the initial version of the web-based system; however, it can easily be scaled for a larger client-base by migrating to a higher-end database running as a standalone server that supports multiple web and application servers.

## **C. SUMMARY**

In this chapter, the requirements for a web-based approach were identified and described in detail. The specific areas covered included the requirements for system users, the requirements specific to the vulnerability process, and the administration requirements that were necessary for a web-based system. Additionally, the technology resources required for developing and supporting a fully web-enabled early warning and

tracking system were identified. Brief descriptions, and selection criteria, were included for the web server, the database server, and the web application development and hosting software used for the final web-based solution. The next chapter will more closely examine the current system implementation and the associated web applications that were developed during this research.

### **III. THE ONLINE COMPLIANCE REPORTING SYSTEM**

The Navy is currently using the Online Compliance Reporting System (OCRS) as an early warning and tracking system for known network vulnerabilities. It was developed in conjunction with this thesis and is a fully web-enabled implementation of the Information Assurance Vulnerability Alert (IAVA) process. The objective for this system is to overcome the problems of data timeliness, accuracy, and accountability that plagued the Navy's original record message traffic based approach. As a result, a variety of functions were converted into web-based applications and made available over the Internet to all Navy commands. Of particular concern were those functions related to Navy-wide notification and individual command compliance reporting. To support these functions in a web-enabled environment also required an extensive array of administrative applications for managing the online user accounts, as well as for maintaining a database containing the Navy's organizational hierarchy.

This chapter begins with a brief description of the interaction between a browser-based client and the various servers that host the web applications. This discussion is intended to provide the reader with a fundamental understanding of how web applications work over the network before examining the inner workings of the actual applications developed for this system. Several of the key vulnerability and administrative web applications are then presented along with discussions on some of the background processing that occurs when they are used. Each application discussed ties back to specific requirements provided in Chapter II, with some of the requirements being restated to better support the discussion.

#### **A. USER INTERACTION WITH WEB-BASED SERVERS**

Before discussing the web applications developed for this system, it will be useful to consider the differences in the flow of data that occurs between a web browser and a web server for static web pages and for web pages that are dynamically generated from a server-based application. The system developed for this thesis is capable of supporting both static and dynamically generated web pages. The server is actually a combination of three different types of servers: a web server, an application server, and a database server. This combination provides for the dynamic page generation capability that is necessary to

support interactive web-based applications. As a result, the web applications and the supporting static web pages are all accessible from the same web server. **Figure 3** depicts the flow of data between a user and the server under both scenarios. Also shown in the diagram, is the domain name, or web page address for accessing the actual web server. Note that it starts with “https,” not “http.” This indicates that an encrypted connection is required between the web browser and the web server. This is one of many security features that will be discussed further in the next chapter.

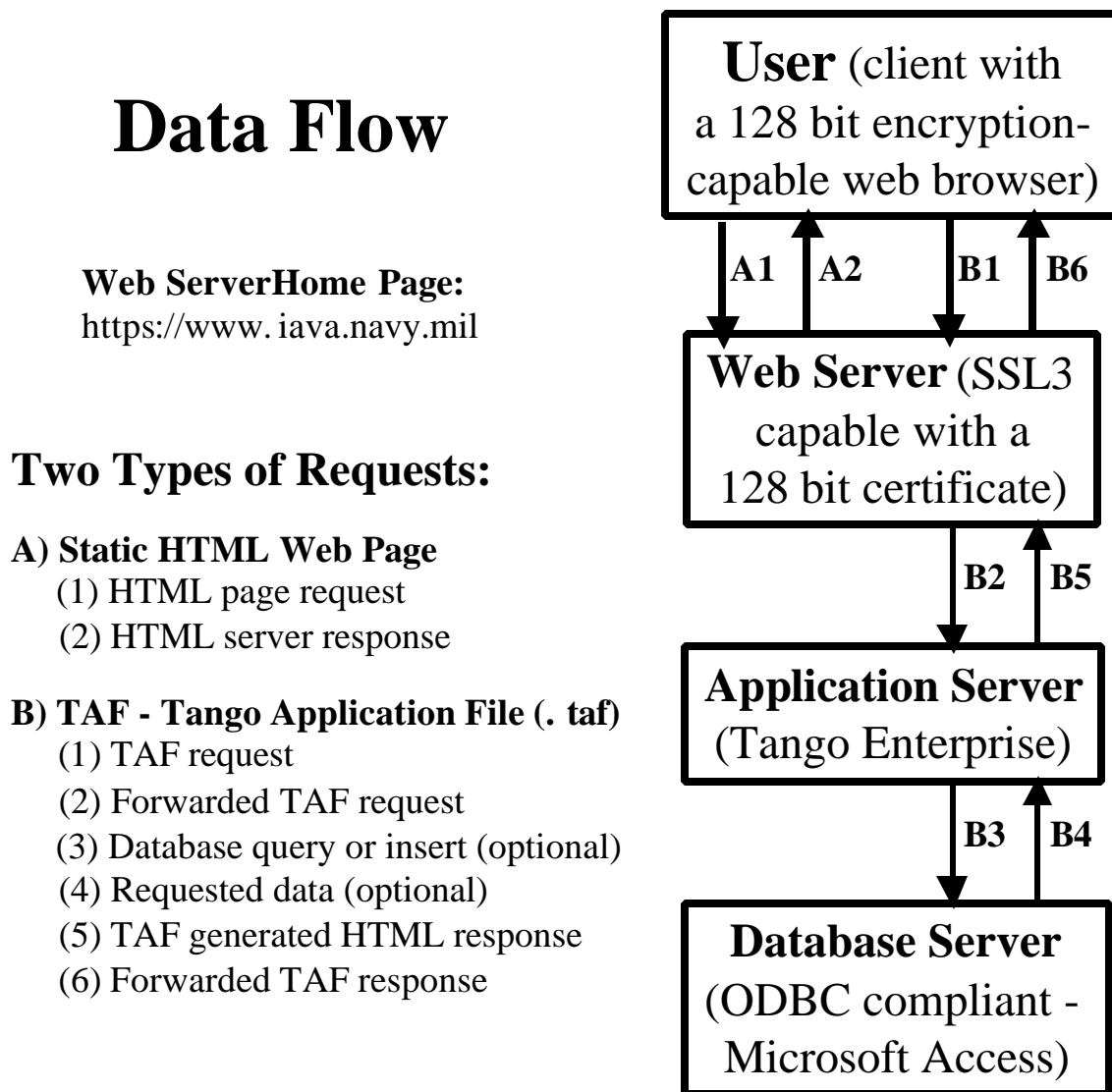


Figure 3. Client-Server Data Flow.

## **1. Requesting Static Web Pages**

In the more common use of web technology, a user, with the aid of a web browser, requests a page from a web server by entering a web page address (or the domain name of the web server). The page, which is stored in a directory on the server, is simply returned to the requesting browser. This is a web transaction that involves a static page, or one that does not change between user requests. **Figure 3** displays the data flows associated with a static page request from a user. The flow labeled A1 indicates a static page request. The flow labeled A2 indicates the web server's response, which includes the complete contents of the web page.

All that is normally required to support a static page transaction is a web browser, a web server, and a connection between the two. However, in this particular case, as **Figure 3** shows, the web server is also required to support the Secure Socket Layer protocol, version three (SSL3), and must also have a 128-bit certificate installed. For the user, the only additional requirement is that the web browser be capable of supporting 128-bit encryption. Almost all web browsers currently available support this requirement.

## **2. Interacting with Web Applications**

One of the capabilities of a web application is that it can dynamically generate a web page based on user input and/or other data available to the application. This capability allows for real-time interaction between a user and a web server that can support a wide variety of information exchange options, especially if a database is used as a repository for the web applications. This is exactly the case with the Online Compliance Reporting System. An application server hosts numerous web applications for administrative and network vulnerability functions. The applications process data coming from a user (via the web server) and occasionally store and retrieve data from a database hosted on the database server. Based on the data processed, the web applications then dynamically generate a web page that is routed back to the user.

A web application is normally triggered when a user requests a web page ending with a unique file extension. In this case, the web server recognizes a request as an application request by the included three-letter extension "taf", which refers to a Tango Application File (TAF). As mentioned in Chapter II, Tango Enterprise provides an

application server and a development environment for creating the web applications that work with the application server. **Figure 3** displays the data flows associated with a web application request from a user. The flow labeled B1 is the original user page request containing the unique “taf” extension. As depicted with the flow labeled B2, the web server recognizes the request as a web application request, and forwards it to the application server for processing. The specific web application requested then processes the user’s request. In doing so, it may need to interact with the database server to insert or retrieve data, which is represented by the optional flows labeled B3 and B4. The web application then generates a response by dynamically constructing a web page, which the application server then sends back to the web server (B5). And finally, as indicated by the flow labeled B6, the web server simply forwards the newly constructed response back to the user.

A user request that only indicates a specific web application is not always sufficient to interact with the application. In many cases, additional data is required from the user to indicate what needs to be done by the application. The user is able to provide this data through the use of web forms that have input fields. In the next section, several key web applications are discussed and some of the associated web forms are presented.

## **B. KEY WEB APPLICATIONS**

Two categories of web applications are integrated into the web-based system’s application server: administrative applications, and vulnerability notification and reporting applications. Administrative applications are necessary to support the desired web-based approach. More specifically, these are used to manage the Navy’s organizational hierarchy and to control access to the system. The vulnerability applications, on the other hand, provide the functional capabilities that are a required part of the Information Assurance Vulnerability Alert process. These applications were developed to meet the objectives for building a web-based system; however, they cannot be deployed via the web without the administrative applications to support them. The remainder of this section will discuss most of the web applications. Those applications that are key to understanding the web-enabled solution will be discussed in more detail.



## 1. Administrative Applications

### a. Authenticating User Access

All but two of the web applications require user authentication to prevent unauthorized access. A login application was developed to support this requirement. The login application is called “Login.taf” and is always the first application visible to a system user (network action officer or system administrator). It is actually a complex application that performs several tasks behind the scenes to better support the user. The web form that is used to interact with the login application is shown in **Figure 4**. From this form, users can either login or access one of the two applications that do not require a login, new account registration and password retrieval, both of which will be briefly discussed later.

## Online Compliance Reporting System



The screenshot shows a web form titled "Login:". Below the title, there are two input fields: "UserName" and "Password". To the right of the "Password" field is a "Login" button. Below the "Login" button, there are two buttons: "Request Account ..." and "Retrieve Password ...".

Figure 4. Login Screen.

When the login application is activated with a username and password, the user is authenticated against the user accounts stored in the database. If the user is not in the database, the login form is sent back to the user with an “Invalid login” message. If the user’s account is in a “pending” status, a web page is sent back to the user indicating that the account has not yet been approved. Also, if the user’s account has a “closed” status, a message informing the user that the account has been closed is returned. Otherwise, the user is authenticated and the application server establishes a session key and several session variables that are used to track the user through the system. To

maintain the user connection and state, the session key is sent back and forth between the user and the server with each subsequent request.

After a successful login, the user will then automatically be taken to one of three other applications (each of which will be discussed briefly in the remaining sections). If the user's password has been marked for expiration (a result of using the password retrieval application), the user will first be required to create a new password. For security reasons, this takes precedence over the other two options. Next, if the user's command has any new subordinate account requests that are pending approval (as determined by checking the database), the user will be presented with a list of those accounts that are pending and asked to process the requests. Otherwise, the user will be presented with a list of the ten most recent vulnerability messages (most recent at the top) that were added to the database. This automated application selection process ensures that critical tasks are always presented to the user on a priority basis and serves to remind the user of pending requirements.

To logout of the system, a user is simply required to click the "Logout" button. This will run the "Logout.taf" application, which will delete the user's session variables on the server and send the user back to the login screen. The "Logout" button is at the right end of the button menu, which appears at the top of every screen in every application after logging in. The button menu is created by the "Menu.taf" application, which is called by all of the other applications in addition to the login application. This approach allows new applications to be added to the menu without requiring modifications to existing applications.

#### ***b. Registering for a New Account***

An application called "Register.taf" is responsible for handling account registration for new users. It is one of the two applications that do not require login access. Registering for a new account could normally be a tedious process for both the requestor and the approver. However, great care was taken during the development of this web application to automate account registration to the greatest extent possible and to minimize the requirements. First, the information required by a registrant was reduced to just enough data to fully identify the user. The objective was to have a relatively clean and inviting web form with as few fields as possible. Furthermore, it was desired to only

require information that was already known to or immediately available to the registrant. By making the process simple and concise, it was hoped that new account registrants would be enticed to use the system as soon as possible without delaying registration due to complex forms and/or difficult to locate information.

The online account registration form is shown in **Figure 5**. It fits on a single screen and includes concise instructions, highlighted in a red font, just above each field. All but two fields are required, the title field and a second phone number field, which many users might not have. The registrant is simply required to enter name, email and phone number data, followed by an organizational identifier. Then the registrant is allowed to make up a user name and password for accessing the system.

## Online Compliance Reporting System

### Request an account:

---

Rank or title (IT1, LT, Mr., Mrs., Dr., etc.). Spelling of First and Last names cannot be corrected later.

Rank:  First:  Last:

---

Your Email Address must be correct in order to be approved for an account.

NIPRNET Email Address:

---

Your COMmercial Phone # is required. Also provide your DSN Phone # if you have one.

Phone # (COM):  Phone # (DSN):

---

Enter your command's 6 digit Unit Identification Code (begins with 'N' for Navy commands).

Command UIC:

---

Create a UserName and Password. Use 8 to 12 characters in each (letters and numbers only).  
The Password must contain at least 2 letters and at least 2 numbers in the total of 8 to 12 characters.

UserName:  Password:  Re-enter:

---

You will be notified via email message if (and when) your account request is approved.  
If any information is inaccurate, the request will be denied and you will need to resubmit.

Figure 5. New Account Registration Form.

The Unit Identification Code (UIC) field is the only piece of information that might not be immediately known to every registrant; however, every Navy activity has one. It is a unique six-digit string that is used to link the user to a Navy command, which must already exist in the database. The code is readily available throughout most

Navy commands. It is also on the military orders for all active duty personnel and is commonly required on administration forms. Using this code instead of the command name allows the system to automatically validate the user's command and prevents errors due to misspelled command names. When the registration form is submitted, a confirmation screen is presented to the registrant with the actual command name, as identified from the database (using the code provided by the registrant). If the code is not in the database, the registrant is informed to first verify the Unit Identification Code entered and then to contact an administrator (network action officer) at the next level in the chain of command to have it added to the system (see the next subsection).

The registration form also describes the specific format requirements for the user name and password fields. These fields are automatically validated when the form is submitted to ensure the format is followed. In fact, when a registration form is submitted, a whole series of checks are performed to validate the request as much as possible. First, all required fields are checked to ensure they were not left blank. Next, the format of the user name and password fields is validated. Then the database is queried against the email address and user name fields to ensure there are no matches with existing accounts. Both of these fields are required to be unique. Requiring a unique email address helps to prevent a user from establishing multiple accounts. And finally, the database is queried for the command name using the code as described above. If the validation process is successful, the user is presented with a confirmation screen to personally verify the request, including the actual command name and the name of the next activity in the chain of command. If validation fails, the registration form is redisplayed with the original information (minus the password for security reasons) and a detailed error message with instructions is included above the form in a bright red font.

If the registrant presses the "Confirm" button on the confirmation screen, the account registration application creates an account in the database for the requestor and assigns it a "pending" status. An email is then automatically generated and sent to the network action officers at the next level in the chain of command to inform them of the pending account request.

*c. Building a Subordinate Organizational Hierarchy*

As mentioned above, a user cannot register for an account unless the user's command is already in the database. However, one of the requirements previously discussed was that no command data could be preloaded into the database. This resulted in the development of the subordinate application, called "Commands.taf", which allows a network action officer with subordinate reporting commands to enter those commands into the database. And, once the commands have been entered, the subordinate network action officers from those commands can then register for an account. The idea was to start this process at the top of the Navy's organizational hierarchy and continue down until every command was in the system. In this manner, there would be a highly accurate accounting of the commands within the organizational hierarchy because all network action officers would have direct control over the list of subordinate commands for which they were responsible.

The image shown in **Figure 6** is the first screen presented to a user when the subordinate application is used. As shown, it provides the information identifying the user's command, which can be edited by the user. Next, the screen lists all network action officers from the user's command. The user can look up personal contact information from the accounts of the other network action officers that are listed; a capability that is provided via an auxiliary application called "ViewAcct.taf". And finally, the screen also lists all of the commands that are subordinate to the user's command. From this section of the screen, the user can add new subordinate commands, or choose to view the same screen for one of the listed subordinate commands. This latter capability allows a user to drill down to the lowest levels of the subordinate organizational hierarchy with relative ease. When directly viewing a subordinate command, a user is also provided with the capability to close the network action officer accounts listed for that subordinate command.

# Detailed Command Information:

**Command:**

UIC: N00063

PLA: COMNAVNETOPSCOM

Last modified by: [LCDR Jacqueline Butler](#)

[View ISIC Details](#)

[Edit Command](#)

**Command Accounts:**

	Name	Phone	DSN	Status
<a href="#">Details</a>	<a href="#">Ms Linda Bushey</a>	(202) 764-0725	764-0725	Active
<a href="#">Details</a>	<a href="#">CDR Mark Harvey</a>	(202)764-2942	764-2942	Active
<a href="#">Details</a>	<a href="#">CWO3 PRESTON GAYMON</a>	(202)764-0099	(312)764-0099	Active

**Command Subordinates:**

	UIC	PLA	Last Mod By
<a href="#">Details</a>	N32858	DCMS WASHINGTON DC	<a href="#">Ms Linda Bushey</a>
<a href="#">Details</a>	N70294	NCTAMS EURCENT NAPLES IT	<a href="#">Ms Linda Bushey</a>
<a href="#">Details</a>	N70272	NCTAMS LANT NORFOLK VA	<a href="#">Ms Linda Bushey</a>
<a href="#">Details</a>	N00950	NCTAMS PAC HONOLULU HI	<a href="#">Ms Linda Bushey</a>

[Add Subordinate](#)

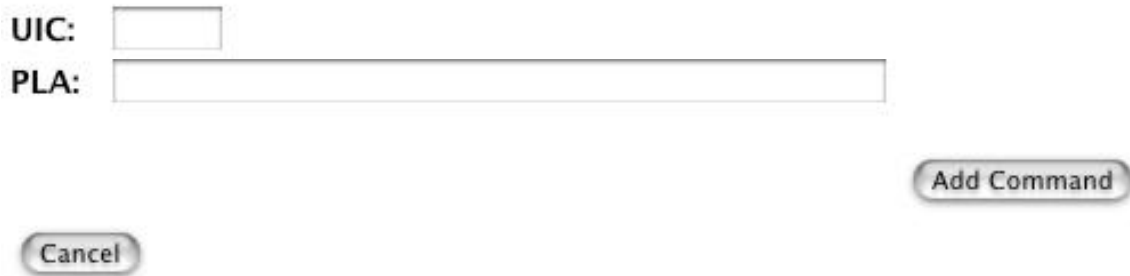
Figure 6. Command Information Screen.

When using the subordinate application to add a new command to the system, only two pieces of information are required, the Unit Identification Code and the Plain Language Address (PLA) for the command, which is simply an official command name. The identification code is required to be unique and is validated as such by the application upon insertion. The subordinate application automatically assigns the

currently viewed command (as seen in **Figure 7**) as the superior command for any command added to the system. In addition, a special rapid indexing field is automatically created and stored in the database with the new command record. This field is used in a variety of ways to efficiently retrieve data from the database for an entire organizational hierarchy using a single query without having to traverse the hierarchy on a query-by-query basis.

## Add a New Command:

- **WARNING:** The UIC must be correct to allow POCs to register.
- **UIC = 6 digit Unit Identification Code (begins with 'N' for Navy Commands).**
- **PLA = Command's Plain Language Address (from SNDL).**



The form contains two input fields: 'UIC:' followed by a small rectangular box, and 'PLA:' followed by a larger rectangular box. Below these fields are two buttons: 'Cancel' on the left and 'Add Command' on the right. The 'Add Command' button is highlighted with a grey border and shadow.

Figure 7. New Command Form.

### *d. Approving Pending Subordinate Account Requests*

Approving accounts is a very critical part of the system's overall security. It is essential that administrators only approve those accounts for authorized subordinate personnel. In a more traditional system, the approval process would likely entail forwarding an account request up the chain of command. Then, an administrator at the top level would verify that the request had been authorized at each level along the way before actually establishing the user's account. This approach is not used because it would be extremely difficult for a small group of administrators at the top level of a large organization like the Navy to personally verify each and every account request. This system takes a completely different approach to new account approval. Instead of recommending approval and forwarding a request up the chain of command, the immediate superior of a user is given the administrative requirement to actually approve

the account in the system. As discussed in Chapter II, there are many advantages to this approach; however, a robust and easy to use application is required to ensure that proper procedures are followed during the account approval process. The account approval application, called “Approve.taf”, was designed to meet these requirements.

The approval form depicted in **Figure 8** is displayed to a user (network action officer at any command) when a pending subordinate account is selected for approval processing. All of the information the user needs to approve or reject the account is available on the screen. There is a checklist, which must be followed. The checklist is part of the approval form that will be processed when the form is submitted to the approval application. Each of the items in the checklist must be checked off. By checking these items and submitting the form, the user is permanently signing his/her name to the requestor’s account as the approval authority, which provides accountability to the process (since it is stored in the database). Also, at the bottom of **Figure 8**, there is a form for rejecting the account request if the current user is unable to properly complete the checklist for the pending subordinate account request.

When a pending account is approved, its status is changed to “active” in the database and the approved user is then able to login. Additionally, an email is automatically sent to the email address of the approved user. The email informs the user that the account is approved and available for use. It also contains some helpful instructions to get the user started using the web-based system.



## Approve a Subordinate Account:

---

**POC:** A Demonstration Request - ([demo@do.not.respond](mailto:demo@do.not.respond))  
**Phone:** (COM) Please Ignore; (DSN) R/ Jim Coffman  
**Command:** N00000 - US NAVY IAVA ADMINISTRATOR

---

### Approve:

#### Approval Checklist:

- ☐ The POC is associated with the Command listed?
- ☐ The POC is authorized to have an account?
- ☐ Each Phone number provided by the POC is correct?
- ☐ The email address provided by the POC is correct?

Approve

### Reject:

#### Rejection Notice: (You may modify the message to provide more detail.)

A Demonstration Request,  
  
Your request for an account has been denied.  
  
LCDR Jim Coffman

- ☐ Reject without emailing above notice.

Reject

Figure 8. Subordinate Account Approval Form.

#### *e. Other Administrative Applications*

There are several additional administrative applications designed based on the system requirements. Each of these web applications is briefly discussed here, but no screen shots are provided. For more specific information on these applications, refer back to the requirements in Chapter II.

The “Account.taf” application allows users to modify personal information within their own accounts. The information that may be changed with this application includes the user’s title (or rank), either of the user’s phone numbers, or the user’s email address. The user must enter a password to make any changes. From within

the “Account.taf” application, a user can access the “Close.taf” application, which simply allows a user to close his/her own account. Once again, the user’s password is required.

The “Move.taf” application is only available to system administrators from the Navy Component Task Force for Computer Network Defense, the host organization. It allows for the reassignment of a command within the organizational hierarchy. In other words, it changes the chain of command. When used, it also automatically changes the chain of command for all subordinate activities of the command being reassigned.

A password retrieval application, called “LostPass.taf” was developed so users could automatically retrieve their passwords if they could not remember them. This is the second of two applications that are available without being logged in to the system. Another password related application is called “Password.taf”. It allows users to change their passwords when desired. It also follows the same password verification rules that the “Register.taf” application does during the account registration process.

## **2. Information Assurance Vulnerability Alert Applications**

### ***a. Early Warning***

The vulnerability process begins with an early warning phase that is intended to notify all network action officers of identified network vulnerabilities. This system implements this capability; however, it does not detect network vulnerabilities. It simply fulfills the information dissemination requirement of the early warning phase. It automates this task by first storing the vulnerability information in a secure online database and then notifying all network action officers (via electronic mail) to login, read the vulnerability message, and report compliance with the protective measures described within the message. The web application that supports this part of the process is called the “IAVAs.taf” application.

One of the capabilities of this application is to post and/or modify vulnerability messages to the online database. This portion of the application is restricted to system administrators. **Figure 9** shows the web form used to post or modify a vulnerability message to the web-based system. Other than a few fields for categorizing the vulnerability, the form is designed to accept the message in its original format. When

this part of the application was first designed, it was intended to automatically create and send emails to all active network action officers in the system database after a vulnerability message was posted using the form. However, despite several different approaches, the Tango Enterprise software has thus far proven incapable of efficiently and reliably handling this task due to the large number of network action officers in the database. As a result, a temporary solution has been implemented until the problems with the automated approach can be fully resolved. The temporary solution employs a separate application, called “Mailing.taf”, which dynamically generates a mailing list of all active users. The mailing list application is also restricted to system administrators, who use the generated mailing list to broadcast the early warning notices. This manual fix only requires a few additional minutes; however, a fully automated mailing capability is still desired.

## Add a New IAVA:

---

**IAVA Number:**

**Short Title:**

**Report Due Date:**  [ mm/dd/yyyy ]

**IAVA Message:**

☐ **Auto-Notify all POCs** (Auto-notification is currently disabled.)

Figure 9. New Vulnerability Form.

After network action officers receive the email notices that follow the posting of vulnerability messages, they can login to the system and read the messages. The web application also supports in this regard by listing all vulnerabilities in the database (most recent first). The network action officers can then select the vulnerability message they are interested in. Then they can either read it, or they can access the compliance reporting and tracking capabilities that are specific to that vulnerability.

***b. Compliance Reporting***

Every command in the Navy is required to submit a compliance report for each vulnerability posted to the system. To support this requirement, a web-based reporting application was developed. It is called “CmdRpt.taf,” and it is accessible from within the “IAVAs.taf” application (mentioned above). This approach allows a report to be linked (automatically by the system) to a specific vulnerability. **Figure 10** shows the web form used by the reporting application to collect compliance report data from a command. As can be seen, the identifying vulnerability information is already included in the header section of the report form. In addition to a specific vulnerability, a compliance report is also required to be associated with a particular command. To ease the reporting process and to prevent inaccurate command information being entered by a network action officer, the command identity is automatically assigned by the system. As shown by **Figure 10**, the command identity is also already included in the header section of the form.

Originally, the reporting application simply assigned the current user’s command identity to a report being submitted by that user. However, based on user feedback, a new requirement forced a change to this approach. For various reasons, some commands could not access the system and instead were required to send their reports to a superior command. The action officers at these superior commands thus required a new capability to submit the reports to the system (by proxy) for their subordinates. This was achieved by allowing the reporting application to instead assign the identity of a selected command vice the user’s command. As a result, any network action officer now has the capability to submit a report for any command in their subordinate hierarchy. Accountability is still maintained because the identity of the network action officer that submits or modifies a report is always recorded with the report. Selecting a subordinate

command (to submit a proxy report) is only available from within the vulnerability status tracking application, which will be discussed next.

## Command Compliance Report:

[Click here to read / print the detailed instructions.](#)

**FROM:** N70294 – NCTAMS EURCENT NAPLES IT

**POC:** LCDR Jim Coffman

**SUBJ:** IAVA #2001-A-5001, CODE RED WORM PATCH RE-VERIFICATION

	Unclassified (NIPRNET)	Classified (SIPRNET)
1. How many assets were affected by the vulnerability?	1a <input type="text"/>	1b <input type="text"/>
2. Of the assets affected in 1, how many have been corrected?	2a <input type="text"/>	2b <input type="text"/>
3. Of the assets not corrected in 2, how many have been granted an extension?	3a <input type="text"/>	3b <input type="text"/>
4. Of the assets not corrected in 2, how many have an extension request pending?	4a <input type="text"/>	4b <input type="text"/>
5. Comments / Justification (optional):	<input type="text"/>	
<div>Reset Submit Report</div>		

Figure 10. Vulnerability Compliance Report Form.

### c. Vulnerability Tracking

The “Status.taf” application allows network action officers to track subordinate vulnerability compliance status. This application automatically summarizes data from all reports up to the command level of the network action officer using the application. In other words, the entire subordinate hierarchy is included in the summary. The user of the application can drill down to any level and see the summary as it pertains to a subordinate command, but cannot go above his/her own organizational level. This approach provides a real-time status at each level in the organizational hierarchy; therefore, depending on the user’s level in the chain of command, a different scope will be presented.

One of the governing principles in developing the status application was to place all of the information that might be needed at the user's fingertips. The screen shot displayed in **Figure 11** is the result of this principle. The information is broken out into four areas. The top of the screen shows the current vulnerability and the current command to which the remainder of the status display applies. The section just below this provides the next two key areas of information. First, there is the required summary of the compliance status (inventories) from the subordinate commands that have already submitted reports. At the top level in the Navy, these are the numbers required by the Department of Defense (DoD) to be reported to the Defense Information Systems Agency (DISA). Next is the total number of required reports, as determined by the number of commands in the subordinate hierarchy. This number is used to calculate the total number of missing subordinate reports, which is highlighted in red. In the bottom section, there is a list of the commands that are immediately subordinate to the current command. The list clearly depicts whether each subordinate has submitted a report. From this section, a user may look at or submit a subordinate command's individual report. The user may also lookup the contact information for all network action officers from any subordinate. And finally, the user may drill down to see the status screen as it applies exclusively to any given subordinate command and its own subordinate hierarchy.

## Status of Compliance Reports

N00000 - US NAVY IAVA ADMINISTRATOR

IAVA # 2001-A-5001: CODE RED WORM PATCH RE-VERIFICATION

Posted: 08/28/01, Modified: 08/28/01

### Summary of Assets (includes all Subordinate Reports):

	Affected	Corrected	Extension Granted	Extension Pending
Unclassified Assets (NIPRNET)	333	324	0	1
Classified Assets (SIPRNET)	315	310	0	1
Totals:	648	634	0	2

Number of Required Reports: 1389

Number of Reports Submitted: 305

Number of Missing Reports as of 09/02/01: 1084

### Status of Immediate Subordinates:

- Press the **Subordinates** button to review the status of that command's subordinates.
- Click on the **Report Date** to view that command's full report.  
If "no report", click on "no report" to Submit a report for that command.
- Click on the **Command PLA** to list the POCs for that command.

	Report Date	UIC	Command PLA
<b>Subordinates</b>	<a href="#">no report</a>	N00014	<a href="#">CNR ARLINGTON VA</a>
<b>Subordinates</b>	<a href="#">08/30/01</a>	N00015	<a href="#">ONI WASHINGTON DC</a>
<b>Subordinates</b>	<a href="#">no report</a>	N00018	<a href="#">BUMED WASHINGTON DC</a>
<b>Subordinates</b>	<a href="#">no report</a>	N00019	<a href="#">COMNAVAIRSYSCOM PATUXENT RIVER MD</a>

Figure 11. Status Tracking Screen.

### C. SUMMARY

This chapter discussed applications developed for the Online Compliance Reporting System and how they worked to accomplish the complete set of vulnerability warning and tracking requirements in a web-based solution. It included a brief

description of the information flow process between the various servers and between the web server and the users when an application is used. Additionally, several of the key applications were discussed, and developer insight into implementation decisions was provided. There is much more to this system than presented here, but the applications discussed provide the best foundation for presentation of this thesis. The next chapter will closely examine the security aspects of system and will then discuss some of the lessons learned during development and use of the system. Proposals for future enhancements will also be discussed.



## **IV. SECURITY, LESSONS LEARNED, AND FUTURE ENHANCEMENTS**

This chapter discusses the security features of the Online Compliance Reporting System (OCRS). Several recommendations for enhanced security capabilities are provided. The security section is followed by a discussion on lessons learned from the use of this system. Recommended solutions to user problems are included with the lessons learned. In addition, the final section details several proposals for future enhancements based on user feedback.

### **A. SECURITY**

There are many features employed in combination to provide a layered approach to security for this system. The objective is to restrict access to persons who are authorized to use it. This includes the protection of information from being read as it is transmitted back and forth across the network. Although the data contained in the online database is not classified, it should be considered sensitive because, like other Department of Defense (DoD) information, it might be possible to abuse it when viewed in conjunction with other sources. Additionally, by maintaining controlled access to the data, it is much easier to ensure the validity of the vulnerability compliance reports received by the system. An integrated Public Key Infrastructure (PKI) approach would address most security concerns; however, the related support technology has not fully matured. The system can be adapted to take advantage of the Public Key Infrastructure in the future. In the interim, a variety of features were used to address security. The remainder of this section discusses the following security features and related concerns:

- Internet domain restrictions
- Encrypted web server connections
- Password format validation
- Ensuring password confidentiality
- Auto-expiring user session credentials
- Account hijacking protection
- Automated password retrieval
- Database security

## **1. Internet Domain Restrictions**

The first security feature encountered by system users is an Internet domain restriction. All clients who access the web server must reside within the “.mil” domain. The web server does a reverse Domain Name Service (DNS) lookup to verify that each client browser resides on a computer from the “.mil” domain (i.e., cs.nps.navy.mil). If the computer being used is from any other domain, it is immediately denied access. Although this may not restrict a determined and technically knowledgeable individual from outside the “.mil” domain, it does serve to generally restrict access to the subset of the Internet population that requires access to the system. Additionally, if necessary, the restriction could be further tightened to allow only users from the “navy.mil” domain.

## **2. Encrypted Web Server Connections**

The next feature employed provides for encrypted connections. The web server uses the Secure Socket Layer (SSL3) protocol to establish encrypted connections with the web browsers used by the network action officers. The server has a 128-bit security certificate and web browsers are required to support 128-bit encryption. As a result, all data that travels across the network between the web server and the web browsers is encrypted. This makes information in transit very difficult to intercept. It helps prevent passwords from being “sniffed” by network-monitoring devices. It also helps prevent analysis of all the vulnerability data being transmitted back and forth.

## **3. Password Format Validation**

To help prevent the use of passwords that can be easily guessed or passwords that may be prone to a common word, or dictionary-style attack, the web applications impose strict password formatting rules. This is intended to save users from themselves by forcing them to make up passwords that are more complex and therefore more secure. Currently, passwords (and user names) are required to be eight to twelve characters long. Additionally, the passwords must be composed of letters and numbers, with at least two of each required. Password security can be further improved with modifications to the “Register.taf” and “Password.taf” applications to impose additional requirements for password formats. For example, a longer password can be mandated, or the inclusion of one or more special characters (like any of these: “!@\$%&\*?”) can be required to further increase the complexity of the password.

#### **4. Ensuring Password Confidentiality**

To help maintain the confidentiality of passwords, online forms that require a password make use of a special field that hides the password from view as it is being entered. This helps prevent “shoulder-surfers” from snatching passwords. Additionally, if a form containing a user’s password is rejected and needs to be redisplayed for data entry corrections, the password is required to be re-entered. This prevents the web server from sending the rejected form back with the user’s password embedded in the web page. Although the password would not be visible from the page when it is redisplayed in the web browser, it would be visible by viewing the source code for that page. This feature also prevents passwords from being stored in a browser’s web cache. And finally, the web applications never format web forms to use the “GET” method when submitting the forms. When the “GET” method is used, it adds each field as a parameter to the web page address. These fields are then visible in the browser’s web page address line when the next page appears. Thus, if a form containing a password were submitted using the “GET” method, the password would be visible. Furthermore, the password would automatically be stored in the browser’s link history file, and possibly even in the bookmarks (or favorites) file. To prevent these security concerns, the web application forms always use the “POST” method, which transmits the form fields to the web server without appending them to the web page address.

#### **5. Auto-Expiring User Session Credentials**

Each time a user logs into the Online Compliance Reporting System, a new session is started on the server to track the user. A unique session key is created and sent back to the user. It is passed back to the server with every page request from the user and is normally stored on the user’s machine in the form of a temporary “browser cookie”. The application server also embeds the session key within every web page sent to the user in the event that “browser cookies” are disabled. The session key is used to uniquely identify the user. It is used to track session variables on the server that maintain the connection state between the user and the server. The session variables are initialized upon login. Without the session key and its associated session variables, a user cannot access the web applications and will automatically be redirected to the login screen for any application request. Every application request checks the user’s session credentials.

This allows the server to differentiate between system administrators and network action officers. As a result, system administrator applications are protected from normal user access. Furthermore, the session variables are automatically deleted after fifteen minutes of user inactivity, which invalidates the session key. This is designed to protect the system if a user walks away from his/her browser for an extended period of time after logging in. The fifteen-minute delay can be reduced if a tighter restriction on an unattended session is desired. It is a configuration parameter on the application server.

## **6. Account Hijacking Protection**

Steps were also taken to protect a user's account when logged in and left unattended prior to the fifteen-minute automatic session expiration. The logged in user's personal information (i.e., email address, etc.) cannot be changed without the user's password. Additionally, the password cannot be changed, nor can the account be closed without entering the user's current password in the required onscreen forms. Taking these precautions prevents a secondary individual from hijacking a user's account when left logged in and unattended; however, it does not prevent a secondary individual from viewing the data in the system and/or submitting or modifying a report. If any data is changed or submitted, it will be tagged (for all to see) with the identity of the network action officer whose account was used. The action officer may then be held accountable for the changes. This provides a good incentive not to leave a connected session unattended. It is possible to prevent unauthorized data entry under these circumstances by requiring modifications to add password fields to all data entry forms. If a password is required to be on all forms, then the web applications that process the forms will also require modifications to authenticate the user when submitted. If these changes were made, a secondary user would not be able to submit or change data, but would be able to view the data already in the system.

## **7. Automated Password Retrieval**

This system provides an automated password retrieval function, as required, to alleviate system administrators of this responsibility. It allows users to retrieve their own passwords in the event they cannot remember them. To do so, a user must enter his/her email address. The "LostPass.taf" application then locates the account with that email address, retrieves the password, and then emails the password to the email address. This

is fairly secure because, even though a secondary individual might be able to enter a user's email address, the password is never displayed. It is always emailed to the user's email address and cannot be retrieved by the secondary individual unless he/she has access to the user's email account. This provides another reason for user's to secure their computers before leaving their desks. Therefore, to hijack a user's password would normally require a secondary individual to be on a computer in the ".mil" domain, to know a user with an account, to know the user's email address, and to have access to the user's email account to retrieve the password.

In a significantly less likely scenario, a more sophisticated hijacker might be successful by "sniffing" packets off the network. This vulnerability exists because the email message containing the password is transmitted "in the clear" to the user. In other words, the email message is not encrypted. To help mitigate the risk associated with this concern, when the "LostPass.taf" application is used to retrieve a password, the password is marked for expiration. As a result, when the user retrieves the password and logs in he/she will be automatically taken directly to the "Password.taf" application and asked to change his/her password due to the potential for compromise resulting from the email message. The user could ignore the request and simply go to another application; however, each time the user logs in, the "Password.taf" application will be automatically activated until the user changes his/her password. It is possible to tighten the security of this approach by modifying "Password.taf" to lock users out of their accounts if they do not change their passwords immediately after the first login following use of the email password retrieval capability.

To completely eliminate the vulnerability associated with email-based password retrieval, the "LostPass.taf" application could be rewritten to provide an online retrieval method based on a series of several unique questions (more is better), with answers only known to the account user. The "Register.taf" and "Password.taf" applications would require modifications to support user entry of the answers that would be needed to authenticate the user during password retrieval. This approach would require careful consideration of the questions used to ensure the answers could not be easily guessed, especially since this would essentially be an alternate method for logging in to the system. Additionally, it would be best to simply allow a user to immediately create a

new password after the question and answer authentication. This would avoid displaying the current password in the user's web browser and most likely also storing it in the browser's cache. The security of this approach would rely totally on the number and quality of the questions and the confidentiality of the answers used to authenticate users.

## **8. Database Security**

The current version of the Online Compliance Reporting System employs a Microsoft Access database to store information, including all user account information. The database is located on the web server, which is physically outside the host firewall security perimeter. If the host machine for the web server were successfully compromised, the intruder would likely have access to the entire database. Destruction of data is not a major concern due to the twice-daily backup procedure followed by system administrators; however, confidentiality of the accounts is a concern. A significantly more secure approach to hosting the database would be to install it on a separate machine and place it behind the firewall. This is not possible using Microsoft Access with the Tango Enterprise software; however, almost any other Open Database Connectivity (ODBC) compliant database can be used. To support this change, the data would have to be loaded into a new database and all of the web applications would have to be redirected to the new data source. Using this approach, only the web server, the application server, and the web applications would be exposed, while all of the data would be secure in the confines of the firewall. Even if the web server's host were compromised it would be extremely difficult to gain access to the database without a valid account, especially since the passwords would also be safely located in the firewall-protected database.

## **B. LESSONS LEARNED FROM USER INTERACTION**

The problems encountered by users of the Online Compliance Reporting System can be generalized into two main categories: training and network access. This section discusses the lessons learned in these areas. According to the system administrators at the host organization, the issues described only affected a very small percentage of the users. Some recommendations for overcoming specific concerns are included in the discussion.

## **1. Training**

The system was put together with ease-of-use as one of the major guiding principles; however, it was soon clear that the author did not accurately anticipate how every single user would react to and interact with the provided interface. The online screens are laid out uniformly across all applications and onscreen instructions are provided where deemed helpful. The onscreen instructions are highlighted using a red font. Onscreen clutter is minimized by using a single menu bar at the top and by including no graphics. Additionally, with each new account approved, the application server automatically sends an email message to the user. The message contains concise, detailed instructions for how to get started.

As reported by the system administrators at the Navy Component Task Force for Computer Network Defense (NCTF-CND), a significant majority of the users have been able to access and use the system with no additional training other than what is automatically provided as described above. However, there have been a small percentage of users who have required help from the system administrators. The system administrators reported that, in almost all cases, had the troubled users simply taken the time to read the email and/or the onscreen instructions (rather than rushing through the screens), they would have been able to do it on their own.

The system administrators have also recently noticed that a small number of users have closed accounts, and then opened new ones. In many of these cases, this occurred because the users had provided incorrect email addresses and then could not remember their passwords. With no way to automatically retrieve their passwords, they simply opened new accounts and then had their old ones closed by a superior. Although there are only a few of these and there is no loss of accountability, this does result in redundant data in the database. Contacting the system administrators could have easily restored the original accounts.

In an effort to better address the above problems and any other common questions from the users, the system administrators are preparing a list of Frequently Asked Questions (FAQs), which will be posted to the web server's home page. Additionally, a monthly system status report will be prepared and sent by email to all active account

holders. The email will also include common and helpful hints for using the system properly.

## **2. Network Access**

The only major technical problem that has been encountered is one in which the web server denies a valid user access to the system. In reality, it is not actually a technical problem with the server, but an implementation policy that results in a denial of access when a certain technical problem exists with the end user's own local network configuration. For security reasons (as mentioned above), Navy policy requires that web servers restrict access to users from the ".mil" domain exclusively. The system's web server follows this policy, which requires the server to do a reverse Domain Name System (DNS) lookup on the Internet Protocol (IP) address associated with each client (user) request. If the lookup does not return a ".mil" domain (e.g., nps.navy.mil) for the host of the client, then the client is not allowed to access the system. There have been quite a few cases where users were unable to gain access because they were not resolving to the ".mil" domain despite actually using a client computer on a ".mil" network. In all cases, the problem was a result of improper network configuration on the user's end. The networks had to be reconfigured to accurately report the ".mil" domain when queried.

There were also some special cases where users simply did not have access to a ".mil" computer. These cases resulted in changes to the web applications to allow network action officers at senior organizations to submit reports for subordinates by proxy. In other cases, users have only had intermittent access. At the Naval Postgraduate School (NPS) for example, access from Spanagel Hall is intermittent, but access from Root Hall or from the dial-in modem bank is always available. The system administrators at the Navy Component Task Force for Computer Network Defense have dealt with this issue extensively and work closely with users to help them correct the problem whenever possible. Some level of intervention from the user organization's network administrator is almost always required.

## **C. PROPOSALS FOR FUTURE SYSTEM ENHANCEMENTS**

This section contains several proposals for enhancing the current capabilities of the Online Compliance Reporting System. They are based on recommendations made by users. Some of the proposals are already in the early planning stages for future addition



to the web application suite. The following five proposed capabilities are currently being considered:

- Additional reporting capabilities
- Acknowledging receipt of compliance requirements
- Consolidated compliance status displays
- Variable user access privileges
- Mandatory periodic password changes

### **1. Additional Reporting Capabilities**

In the author's opinion, the hierarchical administration functions provide the real foundation for the system's overall success. They have been designed to support any type of reporting process for which the desired reports can be specified in a standard format. As a result, the system is especially suited to handle inventory, status, and acknowledgment types of reports, which are very common to the Navy and other Department of Defense (DoD) organizations. The system provides a real-time, online capability with automatic recording of accountability data that includes the organization, the action officer, and the date and time for each report submitted. The reports collected are automatically summarized according to organizational hierarchy; and a drill-down capability is included to allow viewing of each individual report from subordinate activities. Due to the system's flexibility, adding a new reporting capability is fairly straightforward but does require copying and modifying a couple of the existing applications, which can serve as templates. Applications to support a new Tasking Order compliance capability were recently added in this manner (this will be discussed briefly in the next chapter). The effort to do so required only three days. Similar reporting capabilities can be added using the same approach.

### **2. Acknowledging Compliance Requirements**

To further enhance the accountability of the system, the host organization has requested a modification that will allow the system to automatically record a timestamp when a network action officer accesses a vulnerability message that has been posted. The timestamp will serve as an acknowledgement that the vulnerability information has been reviewed and any required action is being taken. Additionally, it is desired that the summary report, as well as the individual reports, reflect current status of these

acknowledgements. These modifications will allow system administrators to track the notification portion of the vulnerability process without having to wait for compliance reports. As time becomes a factor, the system administrators will then be able to proactively notify those organizations that have not accessed a time-critical vulnerability after a certain period of time. This enhancement is considered a top priority and is currently in the initial planning stage.

### **3. Consolidated Compliance Status Displays**

Many users have requested the addition of a single-page, consolidated compliance status report covering their organization and their entire subordinate command hierarchy. As previously shown, the current status report displays an organizational summary followed by the individual status of only those activities that are immediately subordinate to the organization currently being displayed. This is a very concise display, but it still provides access to the more detailed subordinate reports. With the current approach, a user can select any immediate subordinate to drill-down and view the same type of summary report from the subordinate's hierarchical view. The problem is that a user cannot see everything at once (on the same screen). The original prototype contained the consolidated report version, but it was modified to its current form at the request of a Chief of Naval Operations (CNO) N6 staff officer, who felt that the consolidated format would generate too much information (overload) for the higher level organizations. The consolidated status report is currently being planned as an option, but will most likely only be provided after other higher priority requirements. Additionally, a similar consolidated report is being planned, which will allow a user to view their individual organization's status for all network vulnerabilities in a single report rather than having to look them up individually.

### **4. Variable Access Privileges**

Several flag officers and other senior Navy officials have requested the addition of read-only accounts. These accounts are expected to allow for executive-level Navy-wide review of the network vulnerability early warning and tracking process in real-time without having to query system administrators. This is a high priority requirement with no quick and easy solution because there are currently 24 different applications working together on the server and most would require modification, some quite extensively.

These applications must be evaluated from a systemic perspective to determine the extent of any required changes before proceeding.

## **5. Mandatory Periodic Password Changes**

In order to increase system security, system administrators at the host organization have requested automated password expiration capabilities. Currently, a password change is only required after a user activates the password retrieval function, which forces the user to change the password after the next login. There are two additional automatic password expiration capabilities that have been requested. The first is one that would require passwords to be changed after a certain period of time, every 90 days for example. And the second is one that would allow system administrators to force a system-wide password expiration that would require all account users to immediately change their passwords. Both options are being explored by the author and are likely to be added as future enhancements.

## **D. SUMMARY**

In this chapter, the Online Compliance Reporting System security features were described along with some recommendations for increasing the overall security posture of the system. Lessons learned during operational use of the system were also discussed. Plans to help alleviate the primary concerns with user training were included. Additionally, several proposals for future enhancements were provided. The next chapter will address the impact the system has had on the Navy and, in particular, the Information Assurance Vulnerability Alert process.

THIS PAGE INTENTIONALLY LEFT BLANK

## **V. IMPACT ON THE NAVY**

This chapter addresses the impact of the web-enabled Information Assurance Vulnerability Alert (IAVA) process on the Navy. It contains an extensive summary of the Navy's progress with the ongoing operation and use of the Online Compliance Reporting System, a now fully web-enabled early warning and tracking system. Following the summary is a section discussing administrative responsibility and, more importantly, the distribution of that responsibility throughout the Navy's organizational hierarchy. In the author's opinion, this is a factor that has played a critical role in the success achieved with the web-based system. It has greatly impacted the Navy's ability to use the system without requiring additional administrative resources.

### **A. THE NAVY'S PROGRESS USING A WEB-BASED APPROACH**

This section describes the progress and success the Navy has experienced in using the Online Compliance Reporting System as an early warning and tracking system for network vulnerabilities. The following key themes are discussed:

- Navy chooses from two potential web solutions
- Web-based system is available to every command in the Navy
- New system requires minimal administrative support
- System flexibility supports addition of new tasking order capability
- Code Red: a Navy network defense success story
- Web system garners positive feedback and Navy-wide awareness
- Successful system generates interest from other organizations

#### **1. Navy Chooses from Two Potential Web Solutions**

The Defense Information Systems Agency (DISA) developed a web-based system called the Vulnerability Compliance Tracking System (VCTS). In purpose, it is similar to the one developed for this thesis; however, it differs because it is designed from the ground up to be a centrally managed system. As a result, the personnel hosting the system are responsible for administering all user accounts. It is also an inventory-based system. In other words, it requires users to maintain an up-to-date inventory of all network systems. Then, whenever a new vulnerability is posted to the system, it will warn users of the vulnerability, but only those that have the vulnerable systems listed in

their inventories. This is a proactive approach; however, if the inventories are not current, the network action officers may not be properly notified and, as a result, may not respond with the required compliance reports. On the other hand, the system developed for this thesis is a mandatory reporting system. It notifies all network action officers whenever a new vulnerability is posted. Additionally, it requires compliance reports from all commands, even negative reports. The system builds an inventory (of vulnerable systems) based on the reports received. This approach ensures all vulnerable systems are immediately identified, evaluated, and corrected by network action officers when a new vulnerability is posted. It also provides for a more comprehensive and accurate assessment of the actual network systems in use at the time.

DISA offered to host their system as a service to other organizations. This offer was extended to the Navy for an annual cost of \$400,000. At the time, the author had already completed an early prototype of the system based on this thesis. Both systems were demonstrated for staff officers from the Chief of Naval Operations (CNO) N6 office. After carefully weighing the merits and costs, the early prototype, that would ultimately become the Online Compliance Reporting System, was selected for use by the Navy. The Navy Component Task Force for Computer Network Defense (NCTF-CND) hosts and administers the current version of this system, which reportedly costs less than \$15,000 per year to operate. This choice has resulted in annual savings of \$385,000 for the Navy. Also, the host system administrators have reported that they are experiencing close to a 100% success rate for the entire Navy (in terms of the number of systems accurately accounted for with each vulnerability posted). They have also reported that DISA is using their own system internally for vulnerability notification and compliance reporting, but is only experiencing about a 60% success rate for their agency.

## **2. Web-based System is Available to Every Command in the Navy**

As of this writing, there are 1382 Navy commands using the current version of the web-enabled early warning and tracking system. That represents every U.S. Navy activity in the world, with no known exceptions. From these commands, there are approximately 1600 active user accounts (and growing). Many commands have more than one account. There is no limit to the number of accounts allowed per command, but,

for various reasons, some commands have no accounts. Instead, they send their compliance reports to their superiors, who submit them to the system by proxy.

### **3. New System Requires Minimal Administrative Support**

Discussions with the host administrators revealed that there are currently only three personnel who are responsible for managing the Online Compliance Reporting System, and then only as a collateral duty. Between them, during peak access times, they spend no more than two and a half man-hours each workday managing the system. The peak times are rare and tend to occur just before a vulnerability-reporting deadline. The typical daily administrative support time is usually much less than that required during peak times. Administrative duties include performing backups twice a day, monitoring system logs, handling any trouble calls, posting new vulnerabilities, reporting Navy-wide compliance to DISA, and responding to specific queries from Navy leadership regarding Navy-wide compliance status and/or vulnerable system inventories.

### **4. System Flexibility Supports Addition of New Tasking Order Capability**

To date there have been 24 Navy-wide Information Assurance Vulnerability Alerts (network vulnerabilities) posted using the new system. In addition, the web-based system has been recently enhanced to support notification and compliance tracking for Navy-wide Tasking Orders that are issued by the host organization (Navy Component Task Force for Computer Network Defense). Tasking Orders are specific directives that must be responded to by each Navy command. Each Tasking Order requires a compliance report. The Tasking Order enhancement was possible because the administration portion of the system was designed to be flexible enough to handle any type of distributed hierarchical reporting process that can be reduced to a standard reporting format. Since the new Tasking Order capability was added, system administrators have issued six Navy-wide Tasking Orders using the web-based function. Early indications are that this new capability is working just as effectively as the already proven network vulnerability warning and tracking capability.

### **5. Code Red: A Navy Network Defense Success Story**

To illustrate its successful impact, the lead system administrator, Navy Lieutenant Commander Jacqueline Butler, reported that the new web-based system is playing a key role in protecting the Navy's networks. It was used to defend against the recent

worldwide “Code Red” worm attacks that were targeted against Microsoft’s Internet Information Servers (web servers). The Online Compliance Reporting System was used to warn all Navy commands of the potential vulnerability to Microsoft web servers. All commands were required to install the appropriate software patch on their servers and to report compliance status via the web system’s reporting capabilities. The reporting deadline passed before the worldwide “Code Red” attack occurred. As a result of this effort, it was reported that the “Code Red” worm successfully attacked only 7 out of 3200 Microsoft web servers throughout the Navy. Furthermore, the seven servers that were successfully attacked had been reported (using the web system) as being in compliance by their respective host commands; therefore, the threat warning process worked as planned and had the network action officers at those commands succeeded in properly installing the patches, no Navy servers would have been infected.

Without this system, the Navy would not have succeeded in warning the appropriate network action officers, at all commands, of the pending threat, nor would the commands have been able to install the patches and complete the compliance reporting process prior to the “Code Red” attack. Additionally, the Navy would not have had immediate access to the entire Navy-wide inventory of vulnerable Microsoft web servers. The web-based inventory provided a real-time status of the Navy’s posture for defending against the “Code Red” attacks. To further underscore the success of the Navy’s effort, according to Lieutenant Commander Butler, the Air Force and Army fared much worse than the Navy during the attacks despite having the same amount of time to complete the vulnerability notification and compliance reporting process. Both were still using manual methods to implement the process; therefore, they did not have access to real-time statistical and inventory data regarding the numbers, locations, and status of the Microsoft web servers within their organizations.

## **6. Web System Garners Positive Feedback and Navy-wide Awareness**

Overall, the user feedback for the Online Compliance Reporting System has been very positive. In general, the network action officers are extremely pleased with the way they are able to interact with the web-based system. In fact, many have asked to modify the system to support other requirements. Some of the changes requested have already been implemented. For example, network action officers now have the ability to close



subordinate accounts and they can now drill down to manage any level within their subordinate organizational hierarchy. Several other requests are still being considered.

Navy leadership is now keenly aware of the system and its capabilities. Chief of Naval Operations (CNO) N6 staff personnel routinely contact the host system administrators requesting vulnerability compliance status and inventory information. The information provided is being used to proactively ensure Navy-wide vulnerability compliance as well as to measure the effectiveness of the network security policies and procedures within the Navy. As a result of this high level of visibility, flag officers from several senior Navy organizations have requested the addition of root level read-only accounts to provide them with direct access to real-time Navy-wide vulnerability compliance status without having to go through a host system administrator.

## **7. Successful System Generates Interest From Other Organizations**

The Navy is currently the only organization that is collecting and reporting the complete set of compliance data required by the Department of Defense (DoD) for Information Assurance Vulnerability Alerts. This is directly attributable to the successful implementation of the Online Compliance Reporting System. Other organizations are currently reporting little more than percentage compliance estimates (i.e., 80% compliant throughout the Army) based on the manual reports they receive. After seeing the new web-based system successfully used by the Navy and viewing a demonstration, several organizations have expressed interest, including the Army. In addition, the Air Force has decided to build a similar system. Also, the Navy Reserves have expressed interest in adapting it for other purposes. As a result, the system's host command (Navy Component Task Force for Computer Network Defense) is interested in extending the current capabilities. They are planning to budget continued support and development through the Naval Space and Warfare Systems Center in Charleston, South Carolina.

## **B. CRITICAL FACTOR: SHIFTING THE BURDEN OF RESPONSIBILITY**

In the opinion of the author, the ultimate success of the Online Compliance Reporting System has depended heavily on shifting the burden of responsibility for administrative tasks from the central system administrators at the host command to the network action officers throughout the Navy. The action officers at the various levels of the Navy's organizational hierarchy use the system to submit reports for their commands

and, more importantly, to monitor the reporting status of their subordinates. They have a vested interest in actively accepting the administrative responsibilities for managing subordinate user accounts because they are ultimately responsible for the reporting status of their subordinate commands. As such, since their primary objective, with respect to the vulnerability reporting and tracking process, is to get the required information reported up the chain of command as quickly as possible, it is to their advantage to proactively use a system like the one developed for this thesis. However, to do so, they must be provided with access to the administrative functions needed to support their efforts. This web-based system provides them with the automated administration and management tools they need to fully support requirements for the vulnerability notification and reporting process and other similar processes.

By successfully shifting the burden of responsibility from central system administrators to, in this case, over 1600 network action officers, an on-line, web-enabled system provides significant leverage to a distributed hierarchical notification and reporting process for a large organization. In fact, the more administrative functions that can be distributed to all action officers, the more automated the system can become from the end user's perspective while also significantly minimizing the requirement for a dedicated central staff.

### **C. SUMMARY**

This chapter described the progress the Navy has made with the Online Compliance Reporting System, and the successful impact the system has had on the Information Assurance Vulnerability Alert process within the Navy. Additionally, the issue of distributed administration throughout a large hierarchical organization was discussed. It was presented as a critical success factor for the overall impact of the web-enabled solution provided by this thesis. The next chapter will discuss the author's conclusions for this research.

## **VI. CONCLUSIONS**

The successful initiation and continuing operation of the Online Compliance Reporting System (OCRS) has demonstrated that a network vulnerability early warning and tracking capability for a global organization can be achieved using web technology. More importantly, it has shown that distributed and hierarchical administration techniques significantly reduce centralized system support requirements while also maintaining a high degree of accountability and security. The system was designed to overcome the problems of data timeliness, accuracy, and accountability that hampered the Navy's original record message traffic based approach to the Information Assurance Vulnerability Alert (IAVA) process. To achieve this in the context of a web-enabled solution, there were three specific goals: (1) disseminate system vulnerability notices directly to network action officers as quickly as possible; (2) collect compliance reports, then automatically summarize and track the data; and (3) provide a secure online environment for managing the entire process. These goals are tied directly to the system's early warning, tracking, and administration capabilities, respectively.

### **A. WEB-ENABLED EARLY WARNING**

The Navy's original early warning capability was ineffective due to the delays incurred when record message traffic was used to disseminate the warnings. Routing the network vulnerability messages via the Navy's organizational hierarchy, and then through commands to the network action officers, could take several days or more, especially for those at the bottom of a deep hierarchy. This is a significant delay considering the overall 30-day time limit for the entire process and the scope of compliance reporting requirements involved.

The web-based solution derived from this thesis has brought the early warning time down to within a few minutes of the release of a vulnerability-warning message. In fact, network action officers throughout the Navy now receive email warnings before the messages are actually transmitted via the record message traffic network. This is possible because the new system automatically tracks all network action officers that use the system to report vulnerability compliance data. Since each user is required to have a valid email address (which is verified during the account approval process) the system

can dynamically generate a mailing list of all active accounts when a new vulnerability is posted. The mailing list is then used to distribute the early warning emails to all network action officers in the Navy. This rapid information dissemination capability provides a great advantage for the web-based approach. It partially addresses the issue of timeliness and fully accomplishes the first goal of the system.

## **B. WEB-BASED NETWORK VULNERABILITY TRACKING**

The web-based early warning capability leaves more time to accomplish the reporting and tracking of vulnerability compliance data. However, it does not address the delays associated with the reporting process, nor the accuracy or accountability of the data being reported. The reporting delays resulted from the hierarchical roll-up of data (via the chain of command) and were also affected by the efficiency of the slowest reporting subordinate. The inaccurate data resulted from a systemic failure to adhere to a strict reporting format, which could support automatic tabulation. This problem was exacerbated because the record message traffic reporting method could not enforce a format or validate the data prior to submission. The loss of data accountability was also tied to improperly formatted reports, which resulted in guesses as to original intent and led to very generalized status reports that could not be broken down by command.

The Online Compliance Reporting System completely eliminates reporting delays associated with submitting reports via the chain of command. Each report submitted using the system is automatically stored in a database. As a result, real-time status reports are dynamically generated at all levels in the chain of command. The information is available, even at the highest level in the chain of command, immediately after each report is submitted. There is no longer a need to wait for all subordinate reports before forwarding a consolidated report to the next level. In fact, there is no longer a need to forward reports. The system eliminates this requirement by automatically building summary reports based on all reports received. This addresses the data timeliness concerns pertaining to delays introduced by the vulnerability reporting process.

The use of a well-structured form provides the basis for strictly enforcing a reporting format when using the web-based system. This allows a web application to validate the data being submitted before accepting the report. Additionally, if the data is valid, the web application stores a record of the report in the database. This allows other

applications to automatically tabulate the data for dynamic status reports while also maintaining a copy of each individual report received. Furthermore, the identity of the command and of the action officer submitting the report are automatically linked to the report. As a result of this approach, the remaining problems with data accuracy and accountability are addressed and the second goal of the system is fully accomplished.

### **C. DISTRIBUTED ADMINISTRATION FOR A WEB SOLUTION**

The majority of the development effort for the Online Compliance Reporting System was geared towards providing the administration capabilities necessary to ensure the security of the system and the accountability of the information it is collecting. As a result, proper identification and authentication of every user is critical to the success of the web-based solution. Every network action officer requires an account, which allows secure access to the restricted system. Unfortunately, there is minimal administrative support available at the system's host organization to manage the thousands of accounts that are needed for all of the participating commands in the Navy. Thus, a unique approach to managing the accounts is required.

System security is provided using several layers of features to restrict access to authorized users while protecting the data and, more importantly, the accounts used to submit the data (refer to Chapter IV). However, managing the user accounts (registration, account approval, passwords, organizational hierarchy, etc.) needed to access the system presents a significant challenge in light of the limited central support staff and the large number of accounts. To solve this, account administration is entirely decentralized. Administrative functions are distributed down to all account holders in a hierarchical manner. The system maintains the organizational hierarchy and requires network action officers to manage all accounts for users immediately subordinate to them. This has proven highly effective because it evenly distributes the administrative load throughout the entire Navy organizational hierarchy. This distributed administration approach accomplishes the third and final goal for developing a web-enabled solution.

### **D. CONCLUSION**

The Online Compliance Reporting System has successfully addressed the problems with the Navy's initial Information Assurance Vulnerability Alert implementation. As such, it has fulfilled the promise of a web-enabled early warning and

tracking system for network vulnerabilities. Additionally, the distributed administration capabilities that were incorporated into this web-based system have significantly reduced the administrative support requirements for managing a hierarchical information dissemination and collection process. As a result, not only does the system solve the network vulnerability notification and reporting problems, but it also provides an excellent platform for tackling similar concerns in other web-enabled systems. The fundamental ideas and experiences gained can also be used as a model for other applications of Business Process Reengineering (BPR). For example, the web-based distributed administration capabilities provided with this system offer an excellent technical foundation for the Web Enabled Navy (WEN) initiative. This initiative, also known as Task Force Whiskey, is an effort to build a Navy web portal to leverage the capabilities of the Navy and Marine Corps Intranet (NMCI) and the fleet's Information Technology for the 21st Century (IT-21) network.

## BIBLIOGRAPHY

Chou, Chien, "Computer Networks in Communication Survey Research," *IEEE Trans. on Professional Communications*, Vol. 40, No. 3, Sep. 1997.

Goubil-Gambrell, Patricia, "Designing Effective Internet Assignments in Introductory Technical Communications Courses," *IEEE Trans. on Professional Communications*, Vol. 39, No. 4, Dec. 1996.

Hager, Peter J. and Scheiber, H. J., *Designing & Delivering Scientific, Technical, and Managerial Presentations*, Wiley, New York, 1997.

Kroenke, David M., *Database Processing: Fundamentals, Design, and Implementation*, Prentice-Hall, 1995.

Reilly, Colleen A. and L'Eplattenier, Barbara, "Redefining Collaboration through the Creation of World Wide Web Sites," *IEEE Trans. on Professional Communications*, Vol. 39, No. 4, Dec. 1996.

Whetzel, John K., "Integrating the World Wide Web and Database Technology," *AT&T Technical Journal*, March/April 1996.

THIS PAGE INTENTIONALLY LEFT BLANK



## INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center  
Ft. Belvoir, Virginia
2. Dudley Knox Library  
Naval Postgraduate School  
Monterey, California
3. Professor Bert Lundy  
Naval Postgraduate School  
Monterey, California
4. Commander Roy Radcliffe  
Naval Postgraduate School  
Monterey, California
5. Lieutenant Commander Jacqueline Butler  
Navy Component Task Force - Computer Network Defense  
Washington, DC
6. Lieutenant Commander Jim Coffman  
Space and Naval Warfare Systems Center  
Charleston, South Carolina